

BAB III

PEMBAHASAN

3.1. Tinjauan Perusahaan

Tinjauan perusahaan berisi tentang sejarah perusahaan, visi dan misi perusahaan, struktur organisasi serta fungsi dari masing-masing bagian yang ada dalam perusahaan tersebut.

3.1.1. Sejarah Perusahaan

Klinik Vidya Medika merupakan klinik yang bergerak di bidang jasa pelayanan kesehatan. Klinik Vidya Medika didirikan oleh Bapak H. Endi Warhendi S.H pada tanggal 23 Mei 2013 dengan surat Izin No. 503/PPI/54-KP XII/2013. Klinik Vidya Medika beralamat di Jl. Kaum Selatan No. 69 Desa Pasirtelaga Kecamatan Telagasari Kabupaten Karawang. Klinik Vidya Medika melayani pasien selama 24 jam terutama untuk warga di Desa Pasirtelaga.

Pada awalnya Klinik Vidya Medika didirikan karena bapak H. Endi Warhendi S.H melihat warga sekitar yang membutuhkan pelayanan kesehatan harus pergi ke rumah sakit yang lokasinya lumayan jauh , maka dari itulah Bapak H. Endi Warhendi S.H berfikir untuk mendirikan sebuah klinik untuk warga sekitar agar jika warga sekitar membutuhkan pelayanan kesehatan tidak perlu pergi jauh lagi. Seiring berjalanya waktu klinik tersebut berkembang menjadi lebih baik disebabkan oleh semakin banyaknya pasien yang berkunjung. Sejalan dengan upaya pengembangan dan peningkatan mutu pelayanan, Klinik Vidya Medika saat ini sedang berencana untuk memperbesar klinik dan menambah kembali fasilitas guna meningkatkan mutu

serta pelayanan dalam melayani masyarakat.

1. Visi

Visi Klinik Vidya Medika adalah :

”mejadikan Klinik Vidya Medika sebagai pusat layanan kesehatan yang diharapkan semua lapisan masyarakat”.

2. Misi

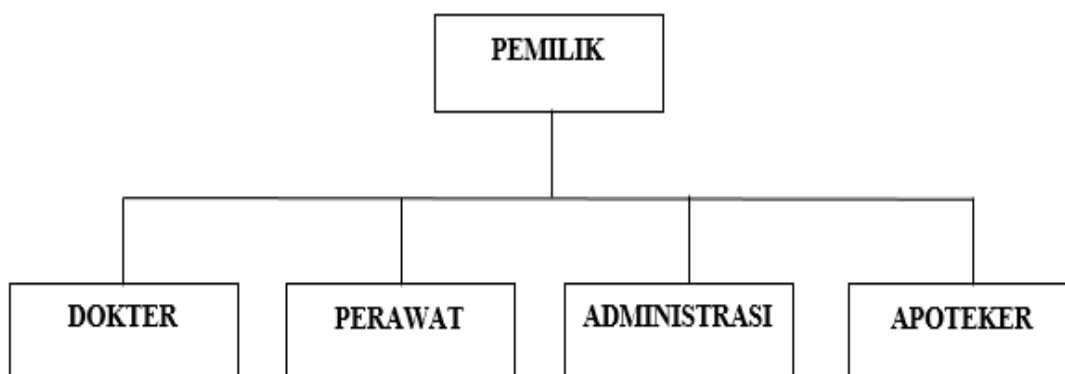
Misi Klinik Vidya Medika adalah :

- a. Menyelenggarakan pelayanan kesehatan yang bermutu, berkualitas, dan profesional dengan mengutamakan keselamatan pasien.
- b. memberikan pelayanan kesehatan yang berorientasi pada kecepatan, ketepatan, keselamatan, dan kenyamanan berlandaskan etika dan profesionalisme.
- c. Mengembangkan sarana dan prasarana yang mengutamakan kualitas pelayanan.

3.1.2. Struktur Organisasi dan Fungsi

Struktur organisasi perusahaan merupakan susunan yang terdiri dari fungsi-fungsi dari hubungan-hubungan yang ada dalam suatu perusahaan yang menyatakan keseluruhan kegiatan untuk mencapai sasaran perusahaan.

Struktur organisasi yang ada di Klinik Vidya Medika dapat di gambarkan sebagai berikut:



Sumber: Klinik Vidya Medika

Gambar III.1
Struktur Organisasi

1. Pemilik

Bertanggung jawab terhadap perencanaan, pengembangan, pelaksanaan, pengawasan, penilaian, dan pengendalian seluruh kegiatan Klinik Vidya Medika yang meliputi kegiatan pelayanan kesehatan, pelayanan keperawatan, pelayanan penunjang medis, serta kegiatan administrasi Klinik Vidya Medika.

2. Dokter

- a. Melakukan pemeriksaan dan pengobatan serta konsultasi spesialis medis pada pasien di klinik.
- b. Memberikan pelayanan rujukan medis serta surat-surat yang berhubungan dengan hasil pemeriksaan kesehatan.
- c. Melaksanakan tugas spesialis lainnya yang di berikan atasan sesuai ketentuan perundangan yang berlaku.

3. Perawat

- a. Bertugas membantu dokter dalam memberikan pelayanan kesehatan dan tindakan khusus yang meliputi wewenang dokter.

- b. Melakukan tindakan keperawatan kepada pasien sesuai kebutuhan dan batas kemampuan.

4. Administrasi

- a. Mengatur segala kegiatan administrasi klinik serta kearsipan klinik.
- b. Melakukan pengecekan terhadap stok obat, data pasien, rekam medis pasien.
- c. Mengelola laporan-laporan yang dibutuhkan klinik secara akurat dan tepat waktu.

5. Apoteker

Membuat racikan obat sesuai dengan resep yang diberikan oleh dokter seperti kegiatan menimbang, mencampur, mengemas dan memberikan etiket pada wadah. Dalam melakukan peracikan obat harus membuat prosedur tetap dengan memperhatikan dosis, jenis dan jumlah obat dengan pertimbangan yang benar.

3.2. Tinjauan Kasus

Tinjauan kasus merupakan penjabaran dalam prosedur kerja yang di terapkan di perusahaan. Dalam proses pelayanan pasien rawat jalan di Klinik Vidya Medika terdapat beberapa prosedur kerja yang berjalan. Prosedur tersebut mengatur agar sistem pendapatan jasa atas pelayanan berjalan sesuai dengan yang di harapkan.

3.2.1. Proses Bisnis Sistem Berjalan

Prosedur sistem berjalan Pendapatan Jasa Atas Pelayanan Pasien Rawat Jalan di Klinik Vidya Medika adalah sebagai berikut:

1. Prosedur Pendaftaran

Pasien yang datang untuk berobat, maka harus melakukan pendaftaran terlebih dahulu di bagian administrasi. Jika pasien baru pertama kali berobat, maka bagian administrasi akan meminta data identitas pasien lalu bagian administrasi akan

mencatat data identitas pasien di buku kunjungan pasien, setelah itu akan membuatkan kartu pasien yang di serahkan kepada pasien tersebut. Setelah itu, bagian administrasi akan menyiapkan bukti rekam medis dan menyerahkan bukti rekam medis kepada perawat dan perawat menerimanya.

2. Prosedur Pemeriksaan pasien

Perawat akan menanyakan keluhan yang di alami oleh pasien dan perawat mencatat keluhan pasien di kartu rekam medis, lalu perawat memberikan kartu rekam medis kepada dokter. Dokter melakukan pemeriksaan kepada pasien, setelah melakukan pemeriksaan dokter akan mencatat hasil pemeriksaan tersebut ke dalam bukti rekam medis dan membuatkan resep obat untuk pasien tersebut.

3. Prosedur Pembayaran

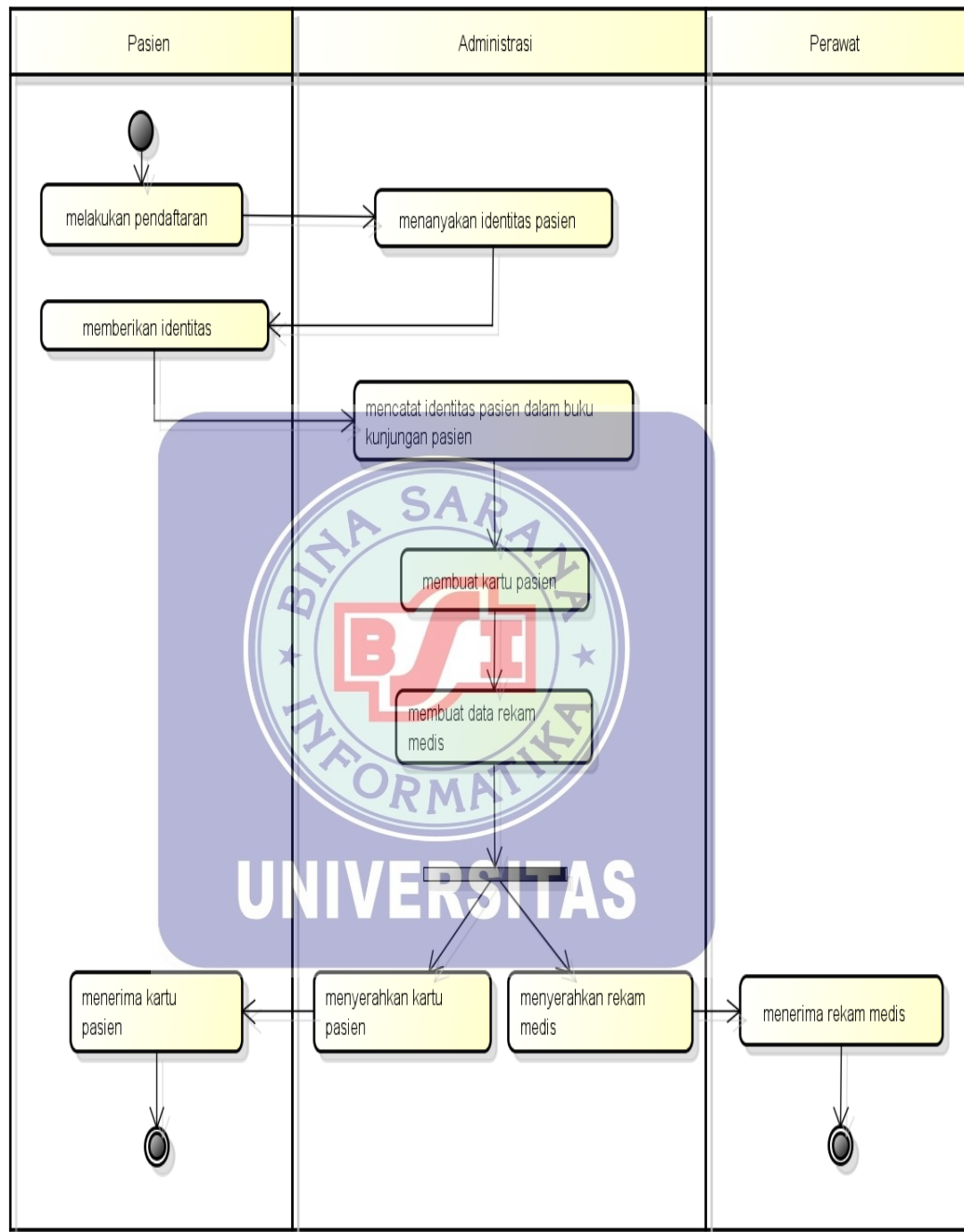
Pasien akan datang ke bagian administrasi untuk melakukan pembayaran dengan menyerahkan resep obat dan bukti rekam medis, lalu bagian administrasi akan menyerahkan resep obat kepada bagian apoteker dan bukti rekam medis akan di arsipkan oleh bagian administrasi. Setelah itu, bagian apoteker akan menyiapkan obat lalu menyerahkan obat ke bagian administrasi, bagian administrasi akan membuatkan kwitansi 2 rangkap, rangkap 1 di berikan kepada pasien dan rangkap 2 di arsipkan oleh bagian administasi. Setelah itu, bagian administrasi akan menyerahkan obat dan kwitansi rangkap 1 kepada pasien lalu pasien melakukan pembayaran.

4. Prosedur Pembuatan Laporan

Setiap selesai melakukan transaksi bagian administrasi akan mencatat transaksi yang ada di dalam kwitansi pembayaran kedalam laporan harian, setelah itu bagian administrasi akan membuatkan laporan bulanan, lalu laporan bulanan tersebut di berikan kepada pemilik Klinik Vidya Medika.

3.2.2. Activity Diagram

1. Prosedur Pendaftaran

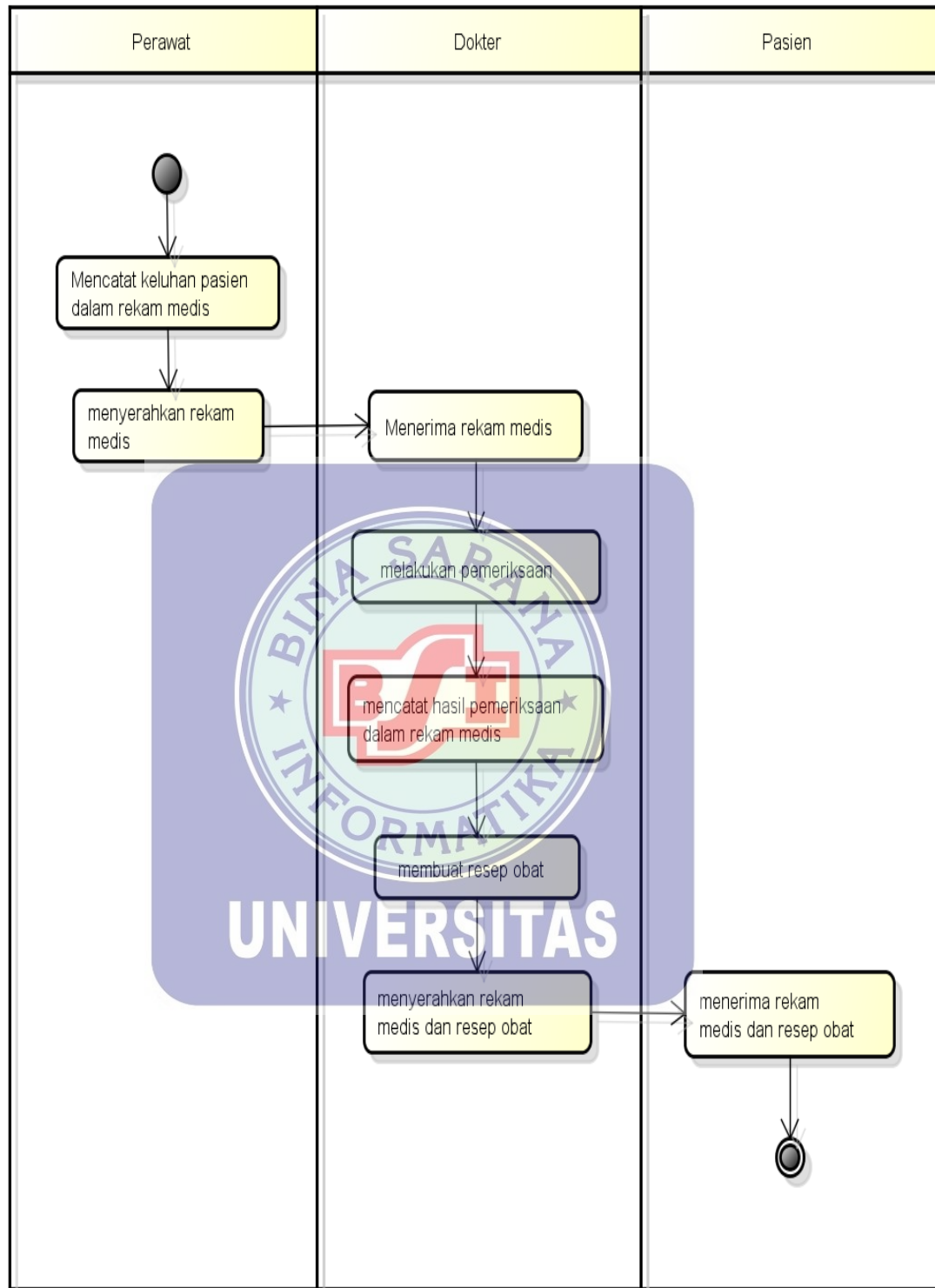


powered by Astah

Sumber : Klinik Vidya Medika 2019

Gambat III.2.
Activity Diagram Prosedur Pendaftaran.

2. Prosedur Pemeriksaan

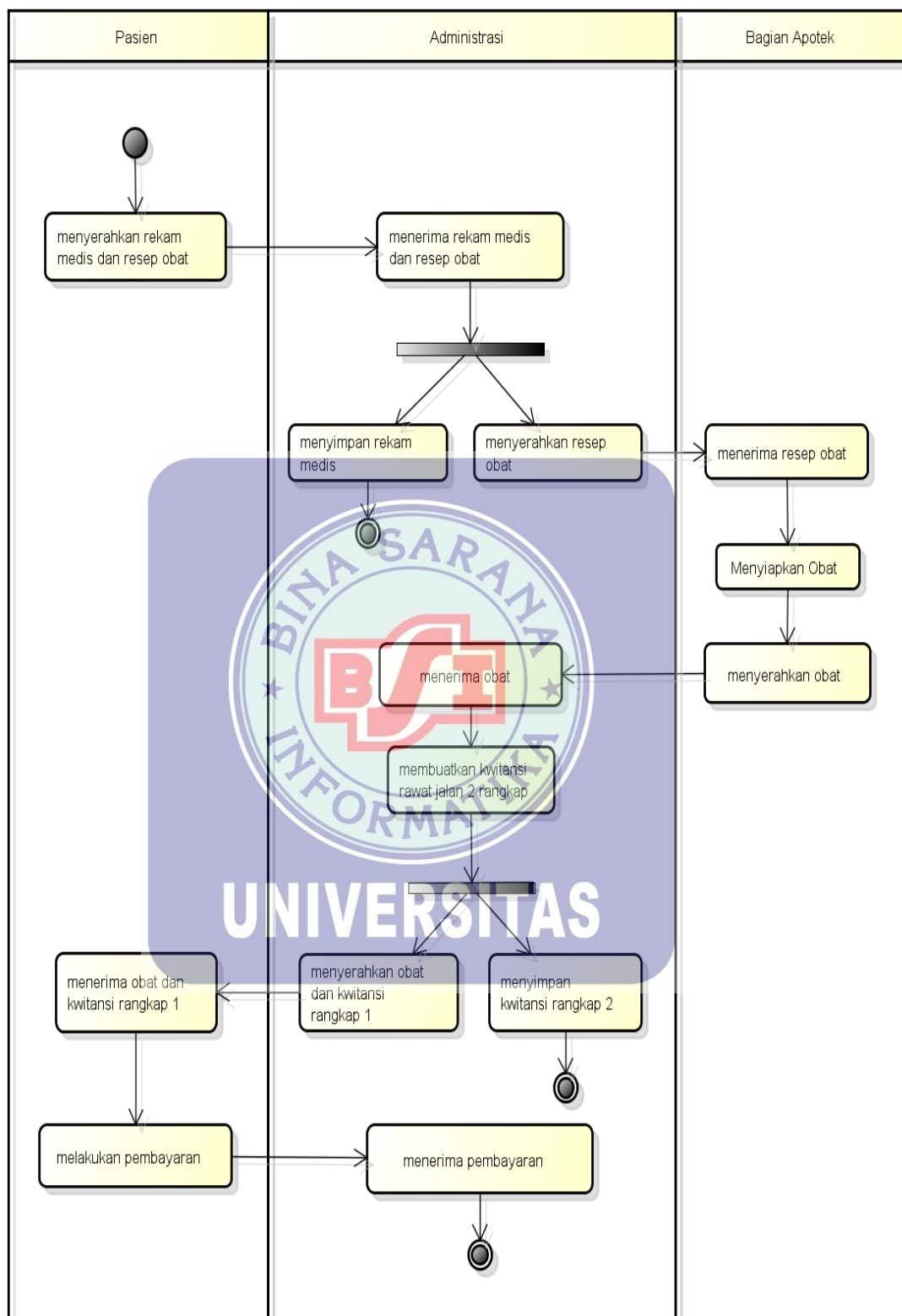


powered by Astah

Sumber : Klinik Vidya Medika 2019

Gambar III.3.
Activity Diagram Prosedur Pemeriksaan

3. Prosedur Pembayaran

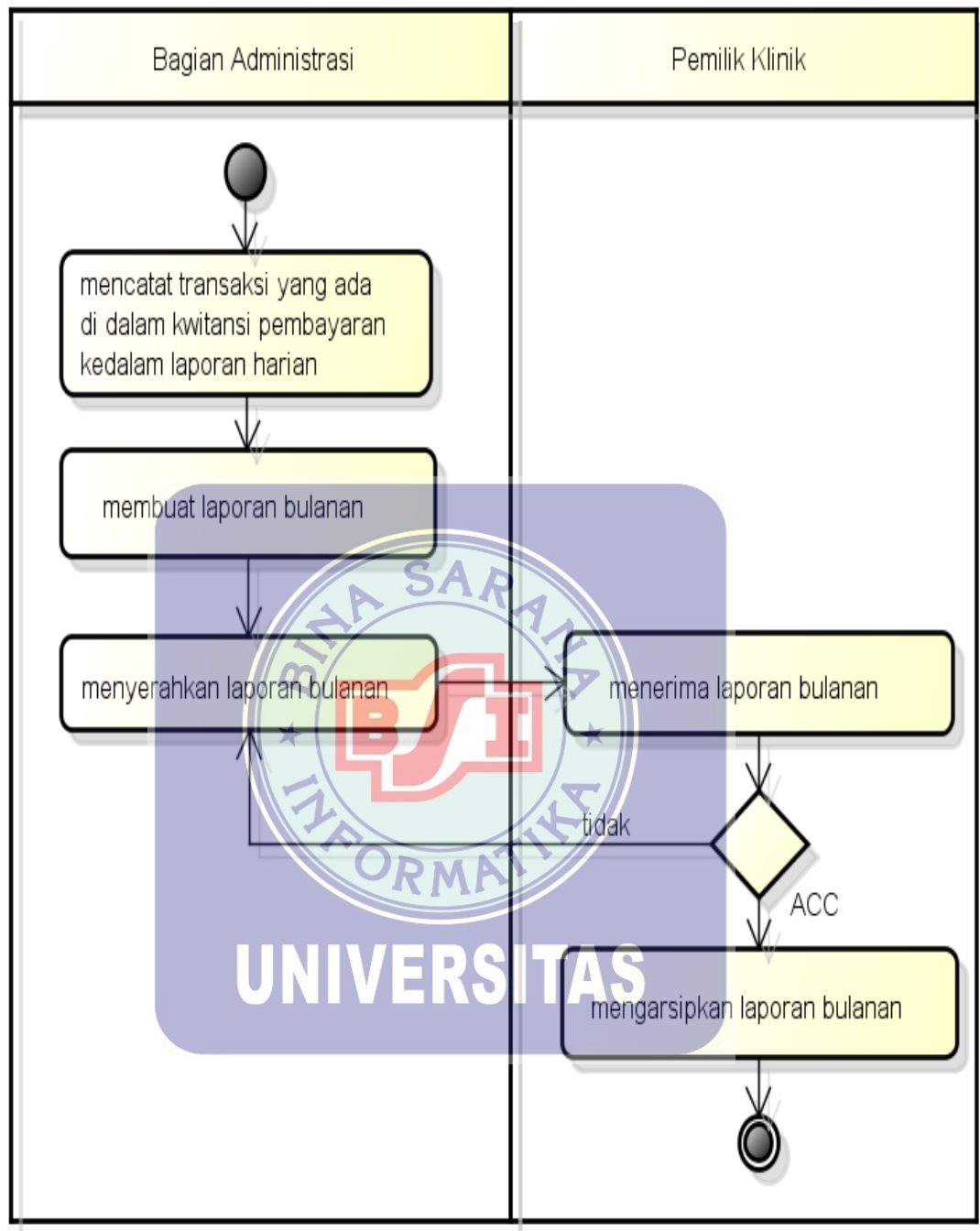


powered by Astah

Sumber : Klinik Vidya Medika 2019

Gambar III.4.
Activity Diagram Prosedur Pembayaran

4. Prosedur Pembuatan Laporan



powered by Astah

Sumber : Klinik Vidya Medika 2019

Gambar III.5.
Activity Diagram Prosedur Pembuatan Laporan

3.2.3. Dokumen Masukan

Dokumen masukan yang ada di Klinik Vidya Medika yaitu :

1. Nama Dokumen : Buku Kunjungan Pasien
 Fungsi : Untuk memperoleh informasi mengenai identitas pasien
 Sumber : Bagian Administrasi
 Tujuan : Arsip
 Media : Kertas
 Jumlah : 1 (satu) lembar
 Format : Lampiran A-1
2. Nama Dokumen : Kartu Rekam Medis
 Fungsi : Untuk memasukan data hasil pemeriksaan selama berobat
 Sumber : Dokter
 Tujuan : Bagian Administrasi
 Media : Kertas
 Jumlah : 1 (satu) lembar
 Format : Lampiran A-2
3. Nama Dokumen : Resep Obat
 Fungsi : Untuk menuliskan resep obat pasien
 Sumber : Dokter
 Tujuan : Bagian Administrasi
 Media : Kertas
 Jumlah : 1 (satu) lembar
 Format : Lampiran A-3
4. Nama Dokumen : Kwitansi Rawat Jalan rangkap 2
 Fungsi : Untuk pembayaran pasien rawat jalan

Sumber	: Bagian Administrasi
Tujuan	: Pemilik
Media	: Kertas
Jumlah	: 1 (satu) lembar
Format	: Lampiran A-4

3.2.4. Dokumen Keluaran

Dokumen keluaran Klinik Vidya Medika yaitu:

1. Nama Dokumen : Kartu Pasien

Fungsi : Bukti pasien telah terdaftar dan tercatat sebagai pasien

Sumber : Bagian Administrasi

Tujuan : Pasien

Media : Kertas

Jumlah : 1 (satu) lembar

Format : Lampiran B-1
2. Nama Dokumen : Kwitansi Rawat Jalan Rangkap 1

Fungsi : Sebagai Bukti Pembayaran

Sumber : Bagian Administrasi

Tujuan : Pasien

Media : Kertas

Jumlah : 1 (satu) lembar

Format : Lampiran B-2
3. Nama Dokumen : Laporan Pendapatan harian

Fungsi : Sebagai bukti laporan pendapatan harian

Sumber : Bagian Administrasi

- Tujuan : Pemilik Klinik
- Media : Kertas
- Jumlah : 1 (satu) lembar
- Format : Lampiran B-3
4. Nama Dokumen : Laporan Pendapatan Klinik
- Fungsi : Sebagai bukti laporan pendapatan bulanan klinik
- Sumber : Bagian Administrasi
- Tujuan : Pemilik Klinik
- Media : Kertas
- Jumlah : 1 (satu) lembar
- Format : Lampiran B-4

3.2.5. Permasalahan Pokok

Dari hasil pengamatan, penulis menemukan beberapa permasalahan-permasalahan yang terjadi pada sistem pendapatan jasa atas pasien rawat jalan di Klinik Vidya Medika yaitu:

1. Pemanfaatan teknologi yang masih belum maksimal seperti penyimpanan data masih di simpan dalam bentuk arsip kertas sehingga tingkat kehilangan dan kerusakan data cukup tinggi.
2. Pencatatan dan perhitungan transaksi atas pembayaran pasien rawat jalan masih dilakukan secara sederhana berupa pencatatan tulis tangan dan dalam pembuatan laporan pendapatan rawat jalan sering terjadi kesalahan karena sulitnya melakukan pengecekan terhadap transaksi atas pembayaran pasien rawat jalan sehingga keakuratan data sangat diragukan.

3.2.6. Pemecahan Masalah

Untuk mengatasi permasalahan, penulis menyarankan beberapa pemecahan masalah pada sistem pelayanan pendapatan jasa atas pasien rawat jalan di klinik vidya medika. Pemecahan masalahnya antara lain.

1. Klinik disarankan menggunakan sistem informasi sehingga memudahkan dalam proses pencatatan dan pengecekan data rawat jalan.
2. Di perlukanya sistem informasi berbasis *desktop* menggunakan bahasa pemograman *java* editor *Netbeans IDE 8.1* dan *database MySQL* agar memudahkan dalam proses pendaftaran pasien sampai dengan proses pembuatan laporan pendapatan jasa atas pelayanan pasien rawat jalan. Sehingga proses akan lebih efektif dan efisien.

3.3. Analisa Kebutuhan Software

Analisa kebutuhan *software* adalah proses menganalisa apa saja yang dibutuhkan perangkat dalam penyusunan sistem usulan. Analisa kebutuhan *software* membagi pengguna berdasarkan kebutuhan yang diperlukan.

3.3.1. Analisisa Kebutuhan

A. Bagian Administrasi

- A.1. Administrasi melakukan login.
- A.2. Administrasi mengelola data Akun.
- A.3. Administrasi mengelola data Dokter.
- A.4. Administrasi mengelola data Obat.
- A.5. Administrasi mengelola data Pasien.
- A.6. Administrasi mengelola data Rekam Medis.
- A.7. Administrasi mengelola data Transaksi Rawat Jalan.

A.8 . Administrasi mengelola data Jurnal.

A.9. Administrasi mengelola data Laporan Jurnal.

A.10. Administrasi mengelola data Laporan Pendapatan.

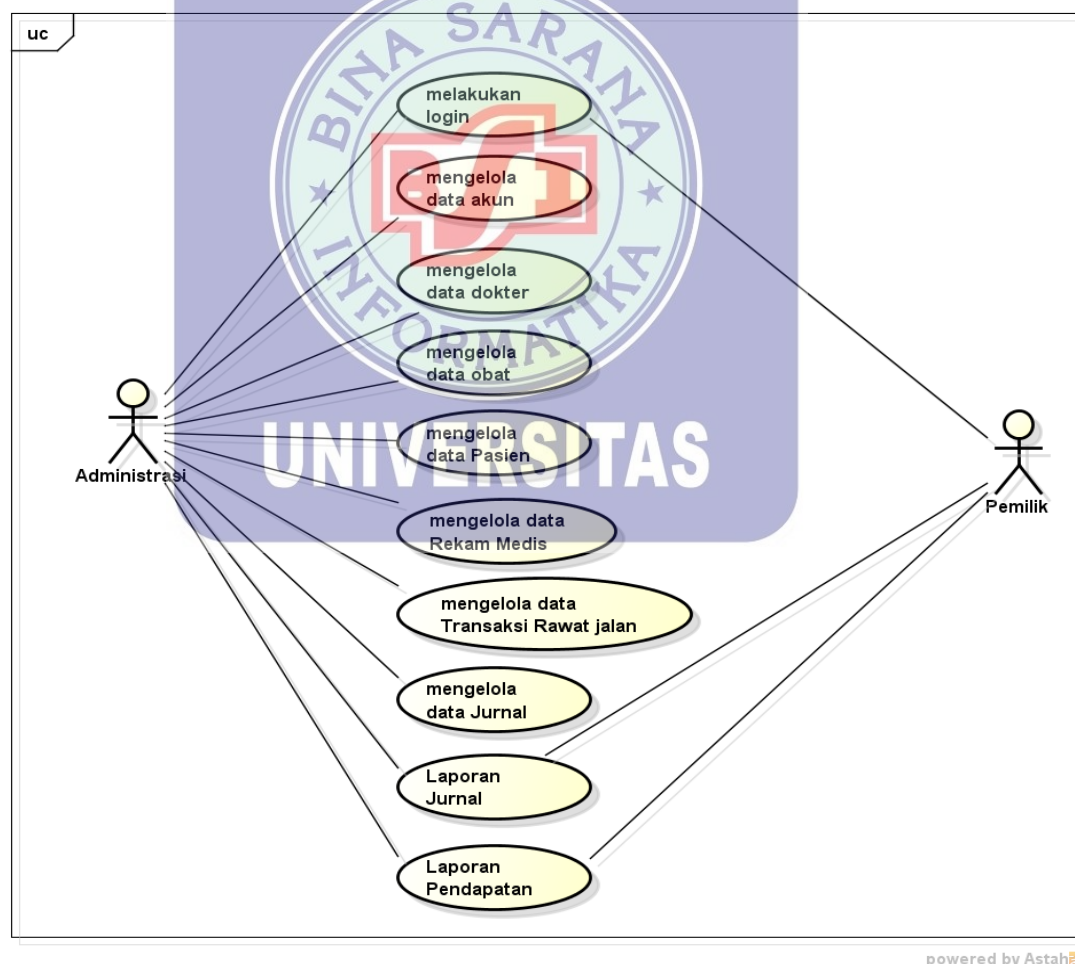
B. Pemilik.

B.1. Pemilik melakukan login.

B.2. Pemilik mengakses data Laporan Jurnal.

B.3. Pemilik mengakses data Laporan Pendapatan.

3.3.2. Use case Diagram



powered by Astah

Gambar III.6
Use Case Diagram sistem usulan

Tabel III.1
Deskripsi Use Case Diagram halaman Administrasi

<i>Use Case Narative Halaman Administrasi</i>	
Tujuan	Administrasi dapat melihat dan mengelola data yang ada didalam sistem halaman Administrasi.
Deskripsi	Sistem ini memungkinkan aktor untuk masuk ke halaman Administrasi yang didalamnya terdapat data akun, data dokter, data obat, data pasien, data rekam medis, transaksi, laporan jurnal, laporan pendapatan.

Skenario Utama	
Aktor	Administrasi
Kondisi awal	Aktor membuka aplikasi halaman Administrasi
Aksi Aktor	Reaksi Sistem
1. Aktor memilih menu data akun	Sistem akan menampilkan informasi menu data akun
2. Aktor memilih menu data dokter	Sistem akan menampilkan informasi menu data dokter
3. Aktor memilih menu data obat	Sistem akan menampilkan informasi menu data obat
4. Aktor memilih menu data pasien	Sistem akan menampilkan informasi menu data pasien
5. Aktor memilih menu data rekam medis	Sistem akan menampilkan informasi menu data rekam medis
6. Aktor memilih menu transaksi	Sistem akan menampilkan informasi menu transaksi
7. Aktor memilih menu jurnal	Sistem akan menampilkan informasi menu jurnal
8. Aktor memilih menu laporan jurnal	Sistem akan menampilkan informasi menu laporan jurnal
9. Aktor memilih menu laporan pendapatan	Sistem akan menampilkan informasi menu laporan pendapatan
Kondisi Akhir	Jika Perintah sesuai maka sistem akan menampilkan sesuai keinginan.

Tabel III.2
Deskripsi Use Case Diagram halaman Pemilik

<i>Use Case Narative Halaman Pemilik</i>	
Tujuan	Pemilik dapat melihat dan mengelola data yang ada didalam sistem halaman Pemilik.

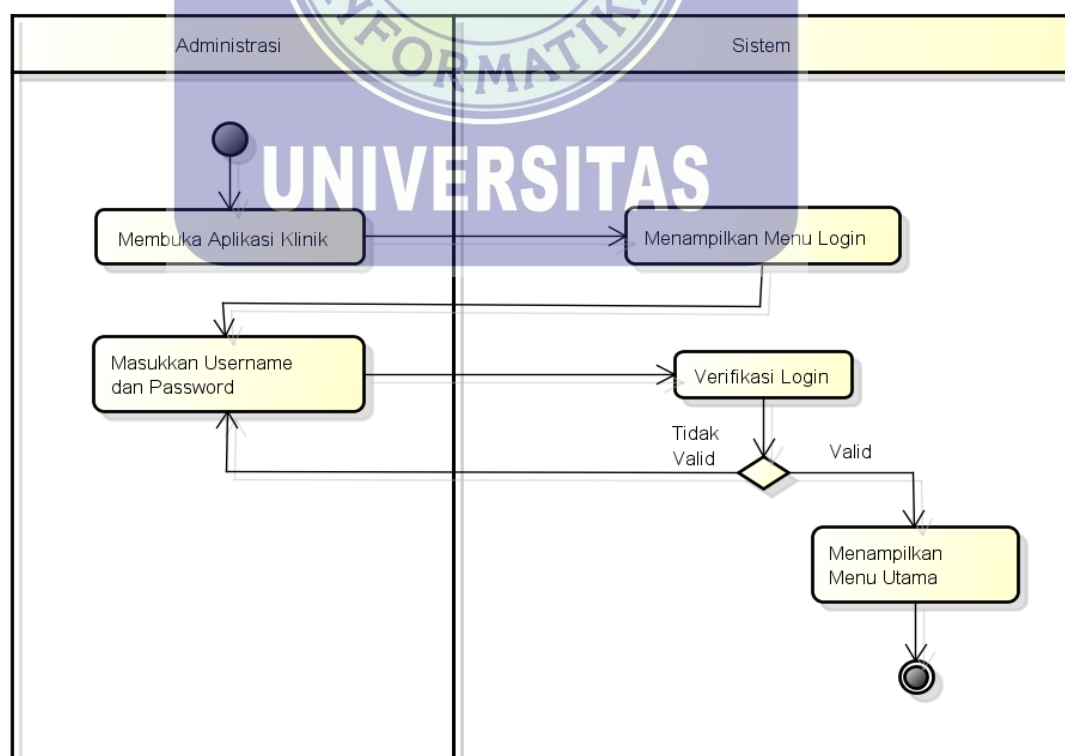
Deskripsi	Sistem ini memungkinkan aktor untuk masuk ke halaman Pemilik yang didalamnya terdapat laporan jurnal, laporan pendapatan.
-----------	---

Skenario Utama	
Aktor	Pemilik
Kondisi awal	Aktor membuka aplikasi halaman Pemilik
Aksi Aktor	Reaksi Sistem
1. Aktor memilih menu laporan jurnal	Sistem akan menampilkan informasi menu laporan jurnal
2. Aktor memilih menu laporan pendapatan	Sistem akan menampilkan informasi menu laporan pendapatan
Kondisi Akhir	Jika Perintah sesuai maka sistem akan menampilkan sesuai keinginan.

3.3.3. Activity Diagram

A. Bagian Administrasi

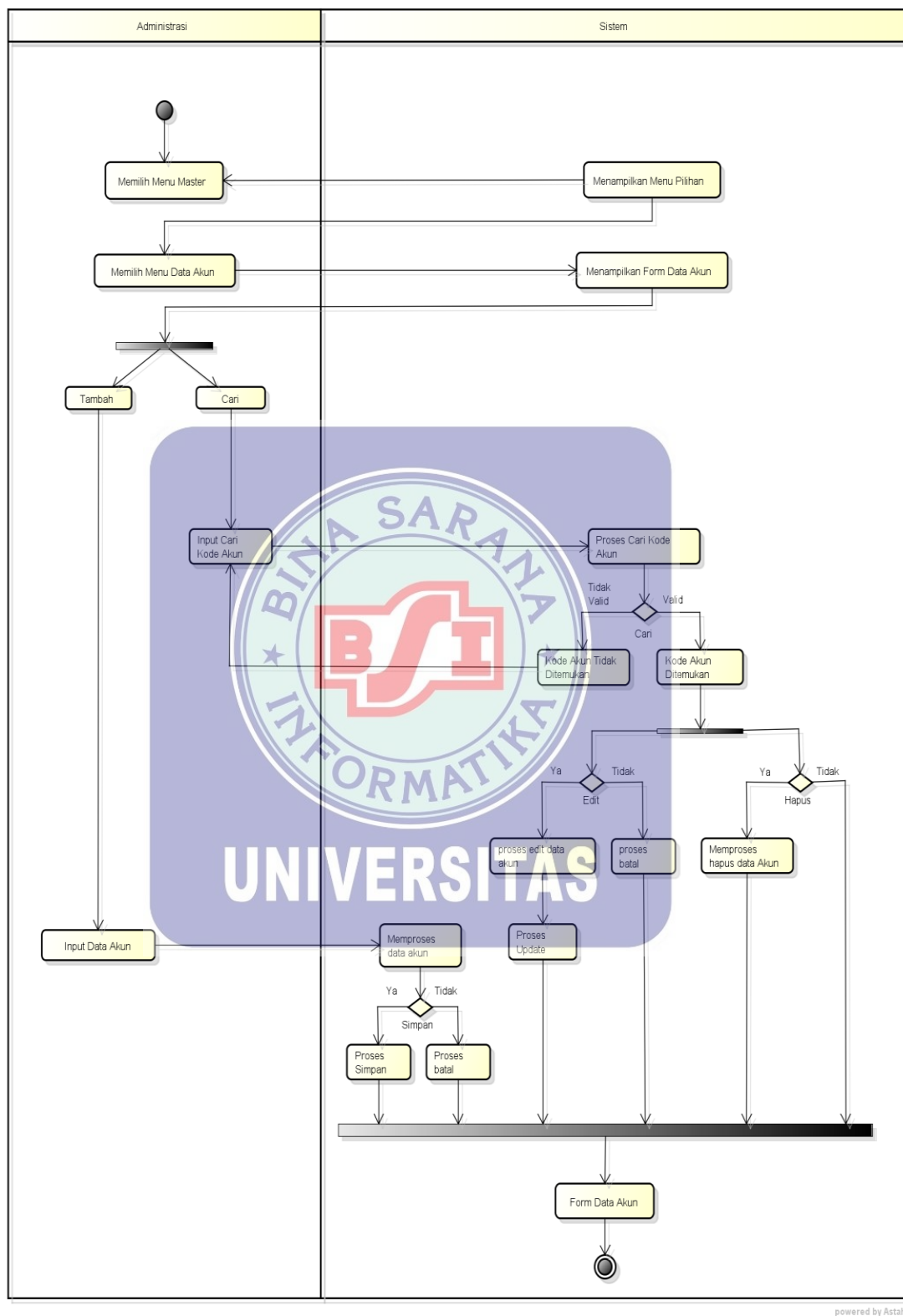
A.1. Activity Diagram Administrasi melakukan Login



powered by Astah

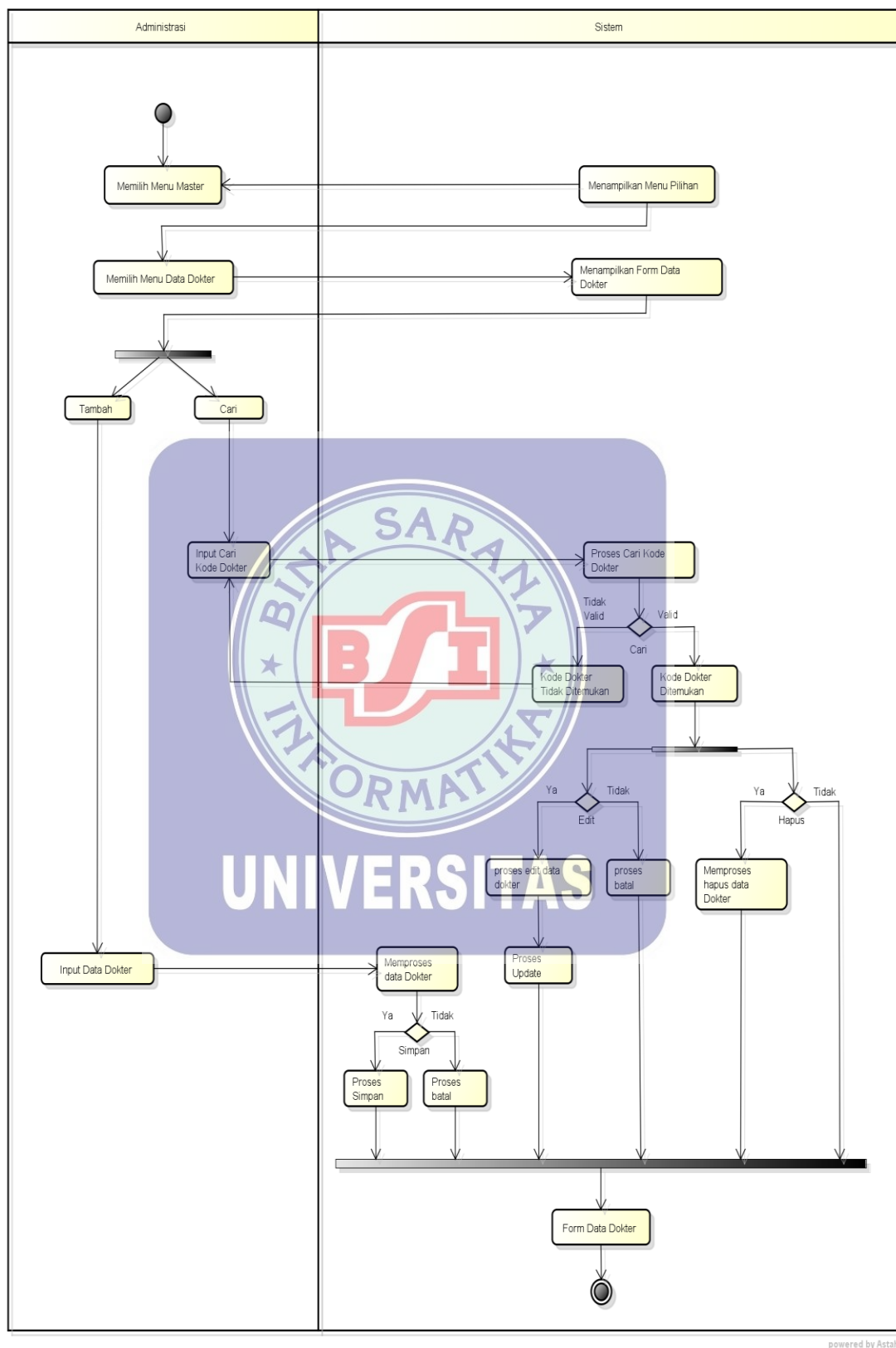
Gambar III.7.
Activity Diagram Administrasi Login

A.2. Activity Diagram Administrasi mengelola Data Akun



Gambar III.8.
Activity Diagram Administrasi mengelola Data Akun

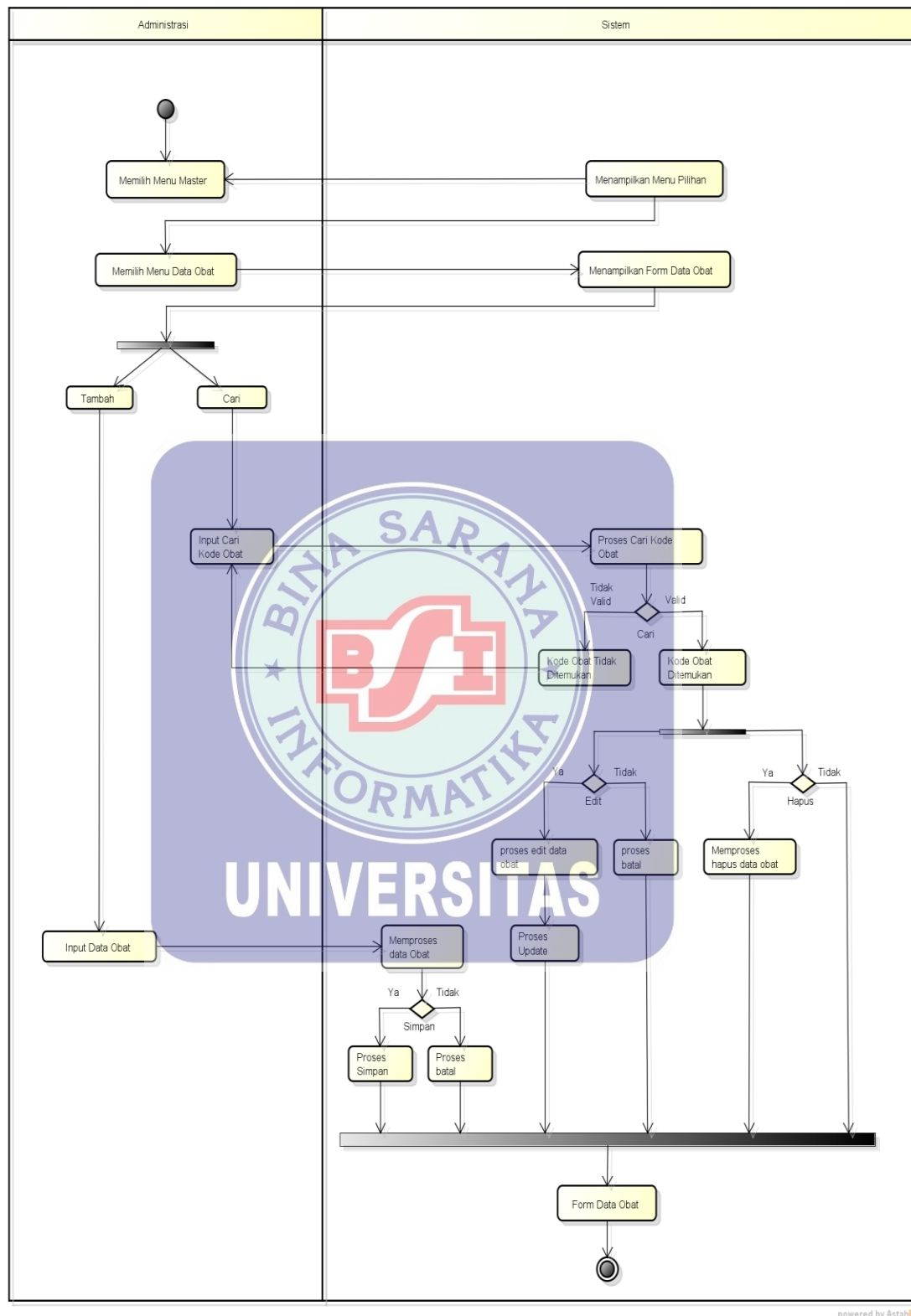
A.3. Activity Diagram Administrasi Mengelola Data Dokter



powered by Astah

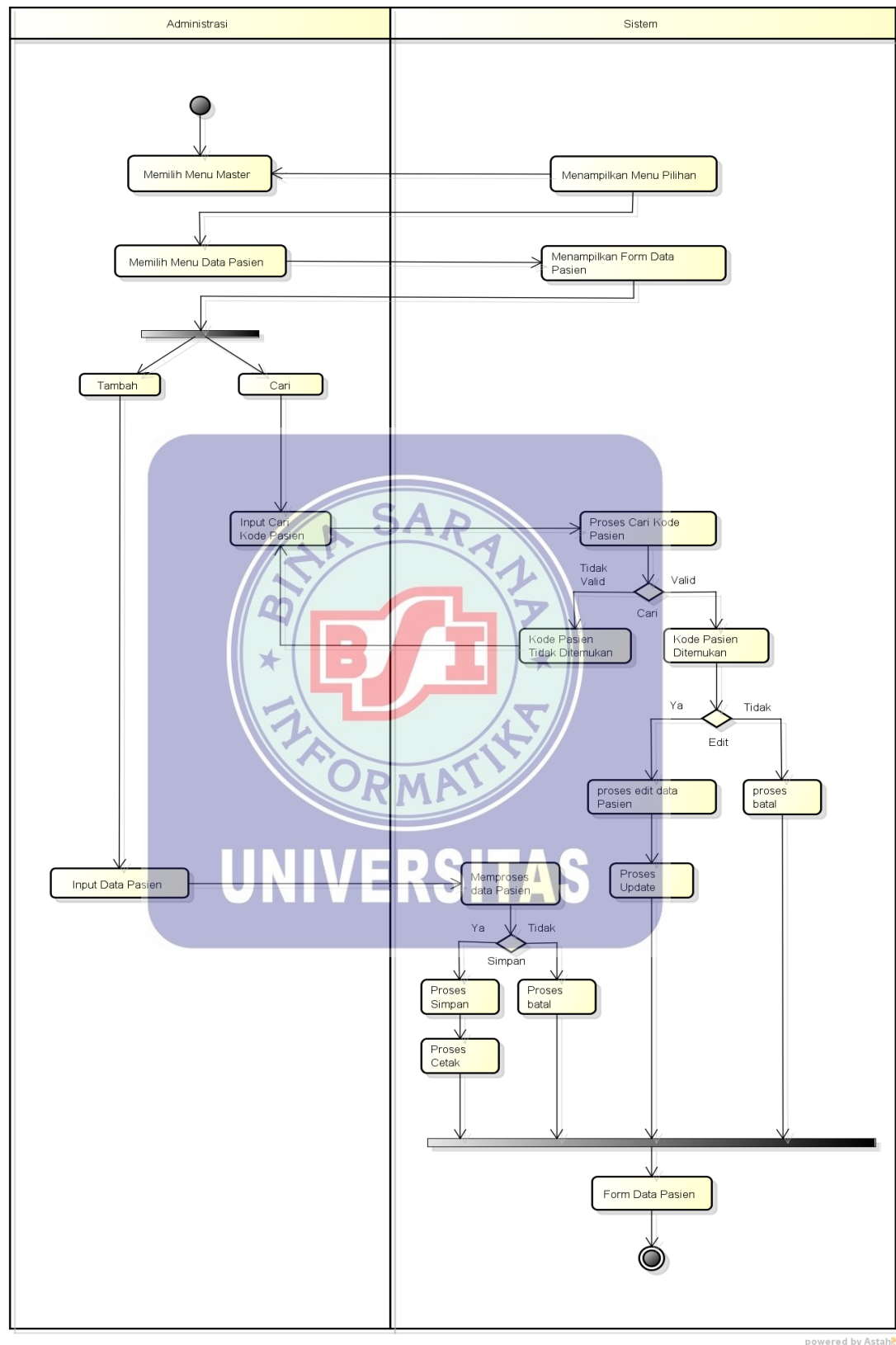
Gambar III.9.
Activity Diagram Administrasi mengelola Data Dokter

A.4. Activity Diagram Administrasi Mengelola Data Obat



Gambar III.10.
Activity Diagram Administrasi mengelola Data Obat

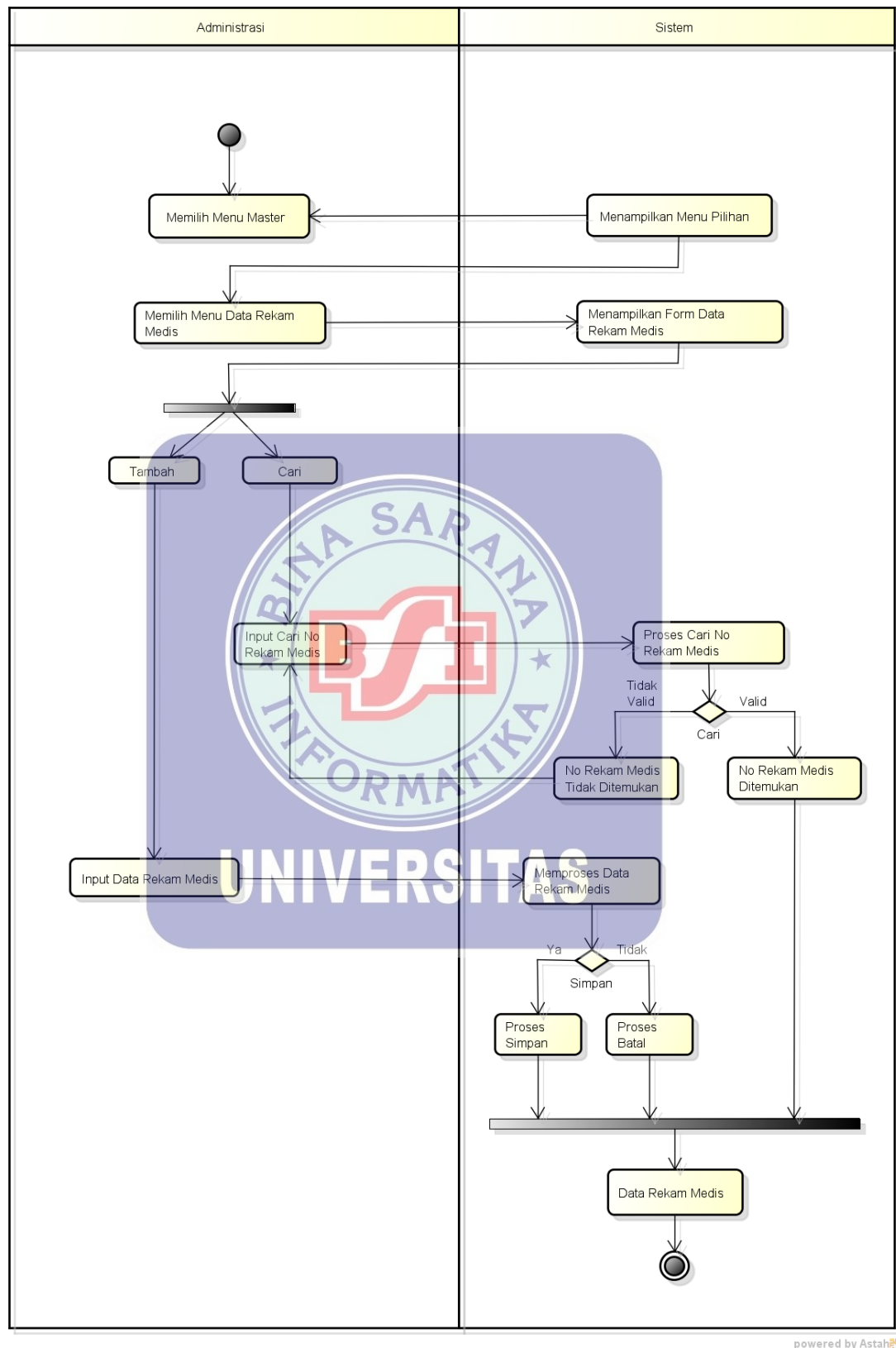
A.5. Activity Diagram Administrasi Mengelola Data Pasien



powered by Astah

Gambar III.11.
Activity Diagram Administrasi mengelola Data Pasien

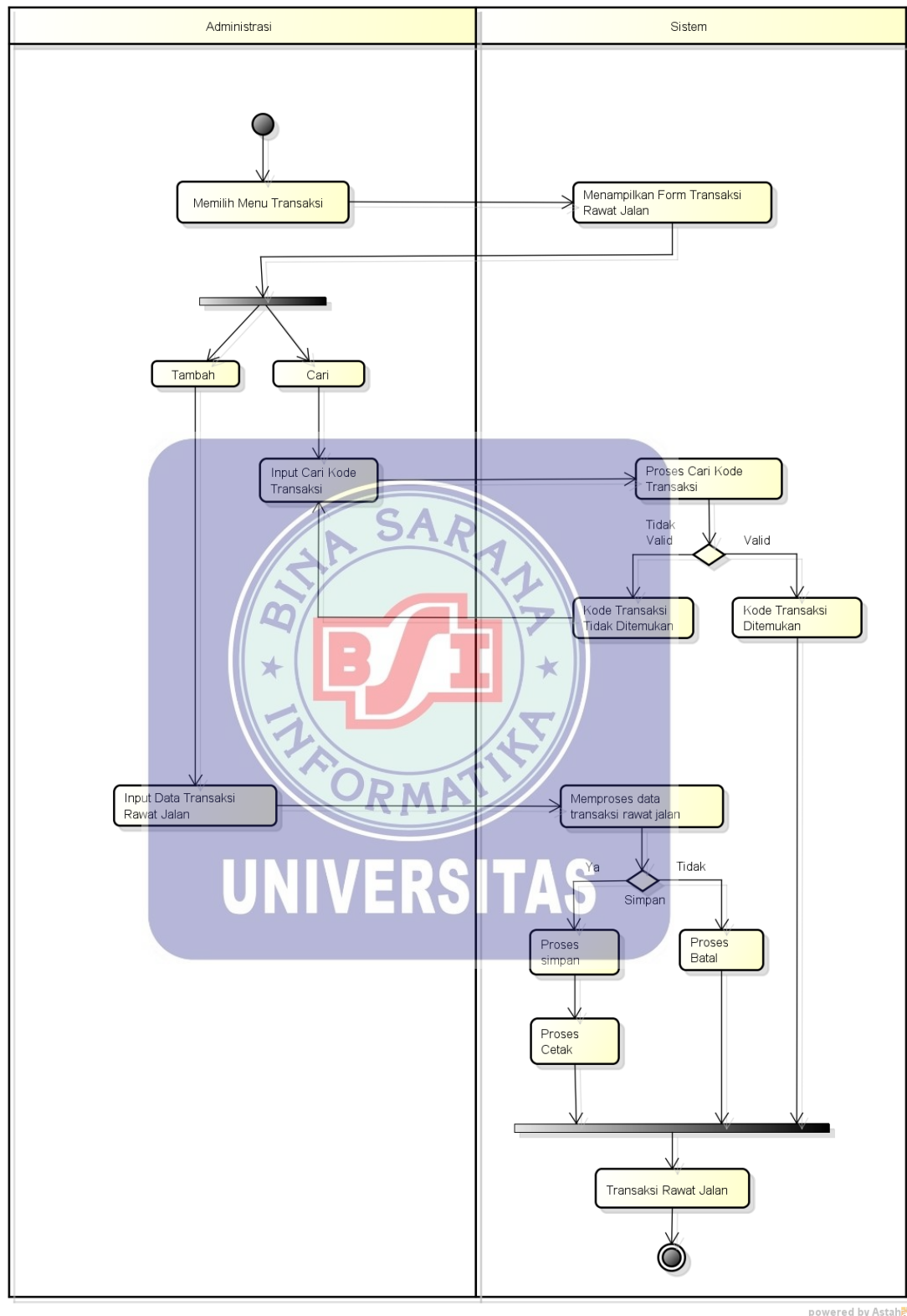
A.6. Activity Diagram Administrasi Mengelola Data Rekam Medis



powered by Astah

Gambar III.12.
Activity Diagram Administrasi mengelola Data Rekam Medis

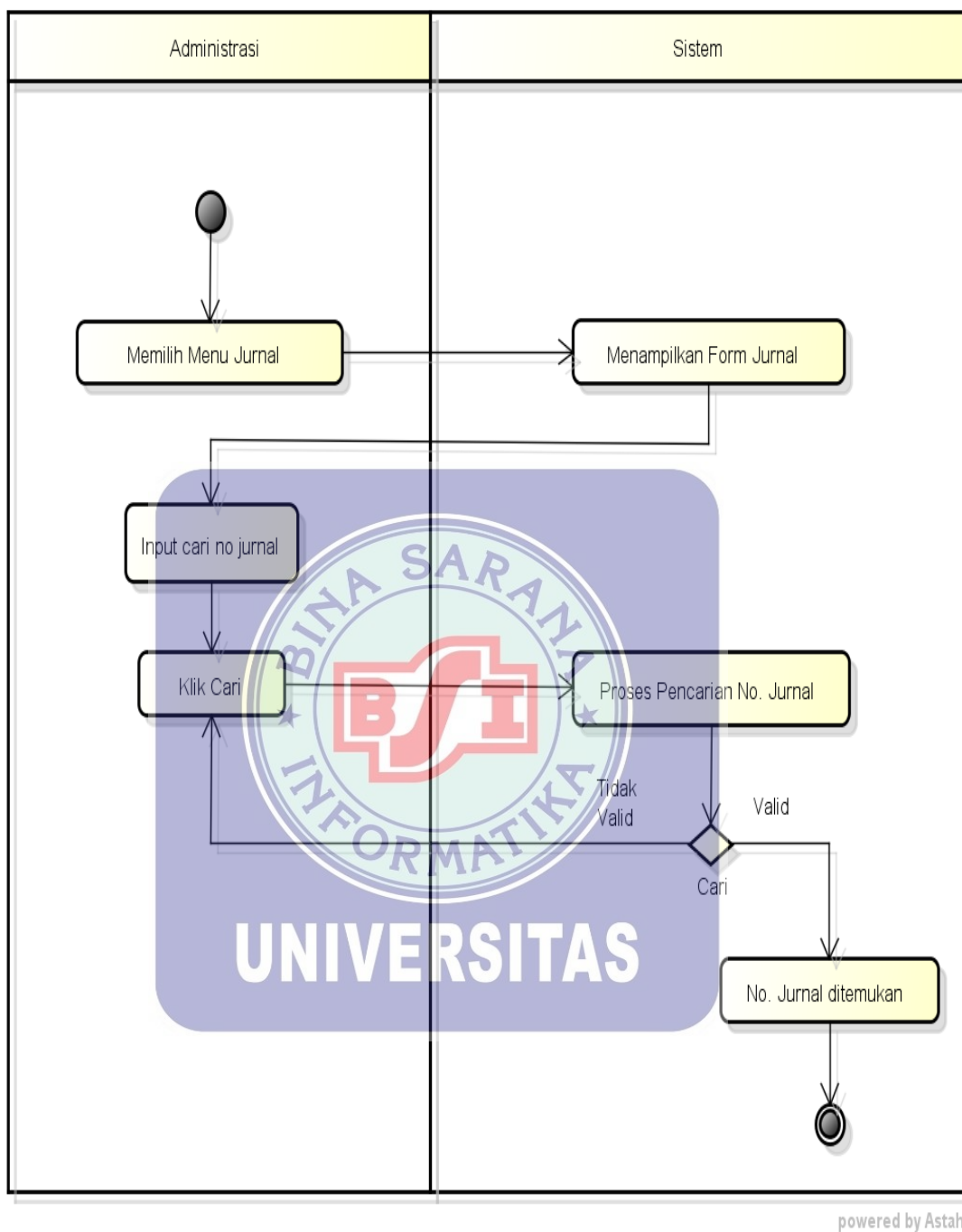
A.7. Activity Diagram Administrasi Mengelola Data Transaksi Rawat Jalan



powered by Astah

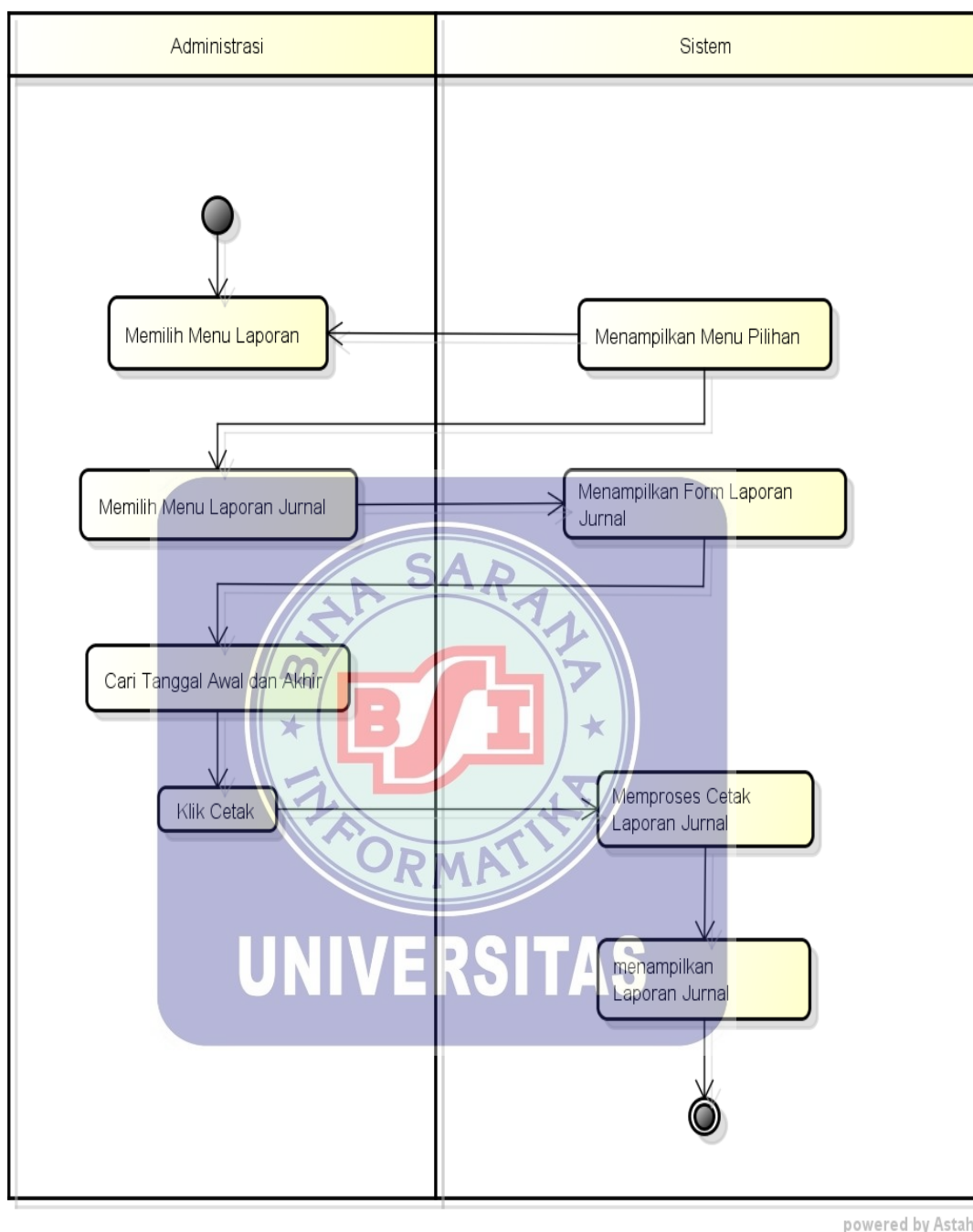
Gambar III.13.
Activity Diagram Administrasi mengelola Transaksi Rawat Jalan

A.8. Activity Diagram Administrasi Mengelola Data Jurnal



Gambar III.14.
Activity Diagram Administrasi mengelola Data Jurnal

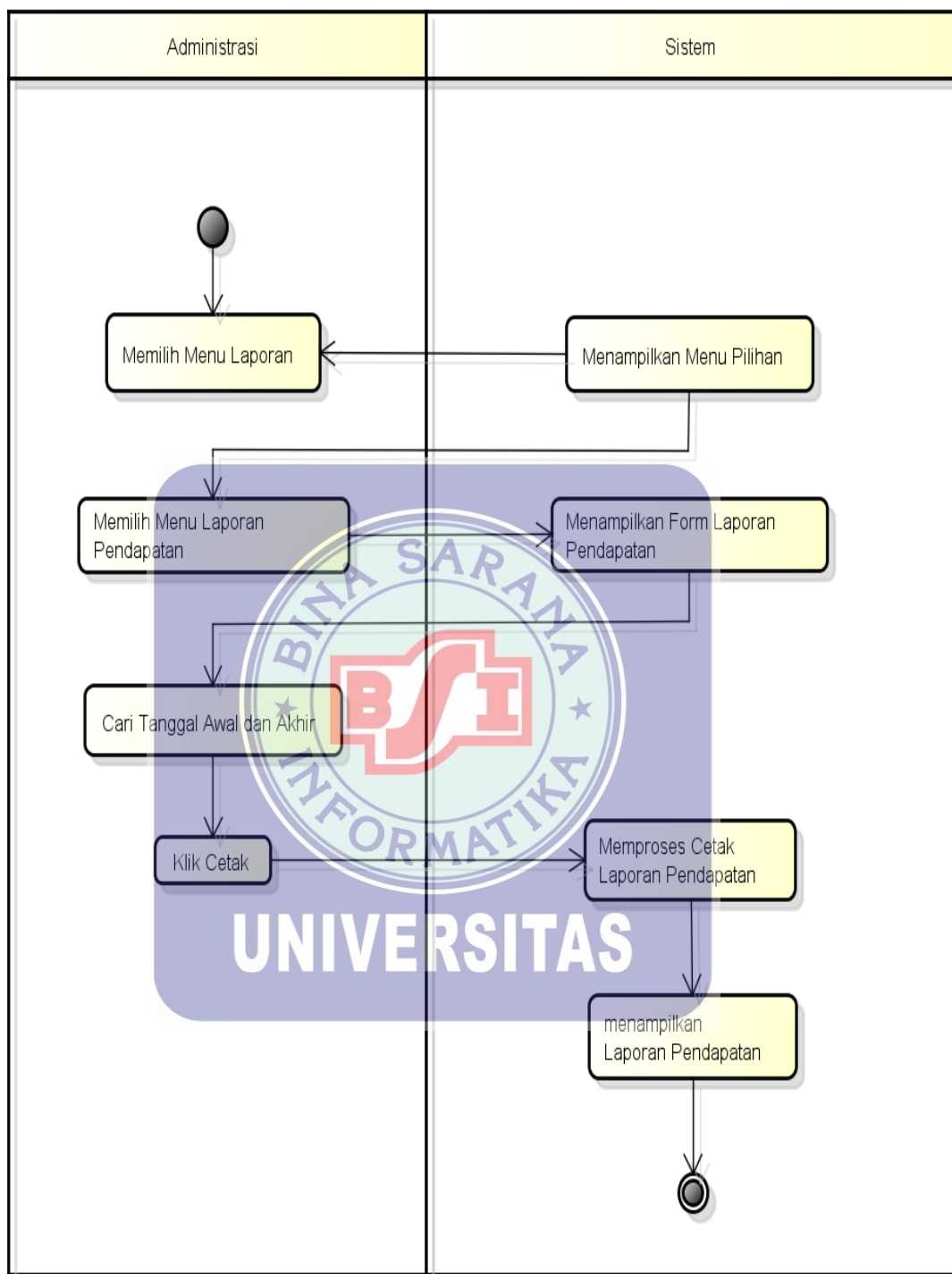
A.9. Activity Diagram Administrasi Mengelola Data Laporan Jurnal



powered by Astah

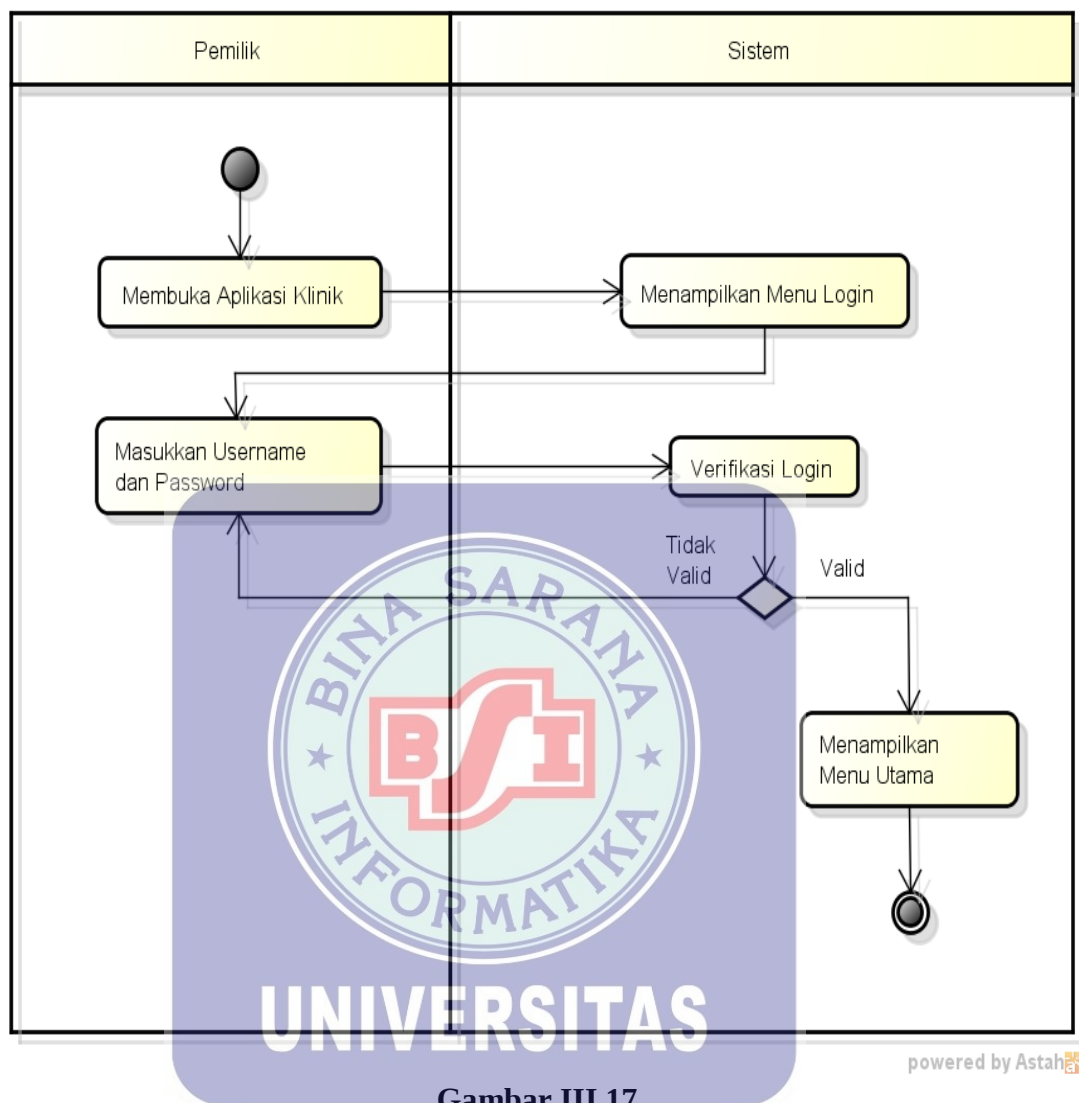
Gambar III.15.
Activity Diagram Administrasi mengelola Laporan Jurnal

A.10. Activity Diagram Administrasi Mengelola Data Laporan Pendapatan

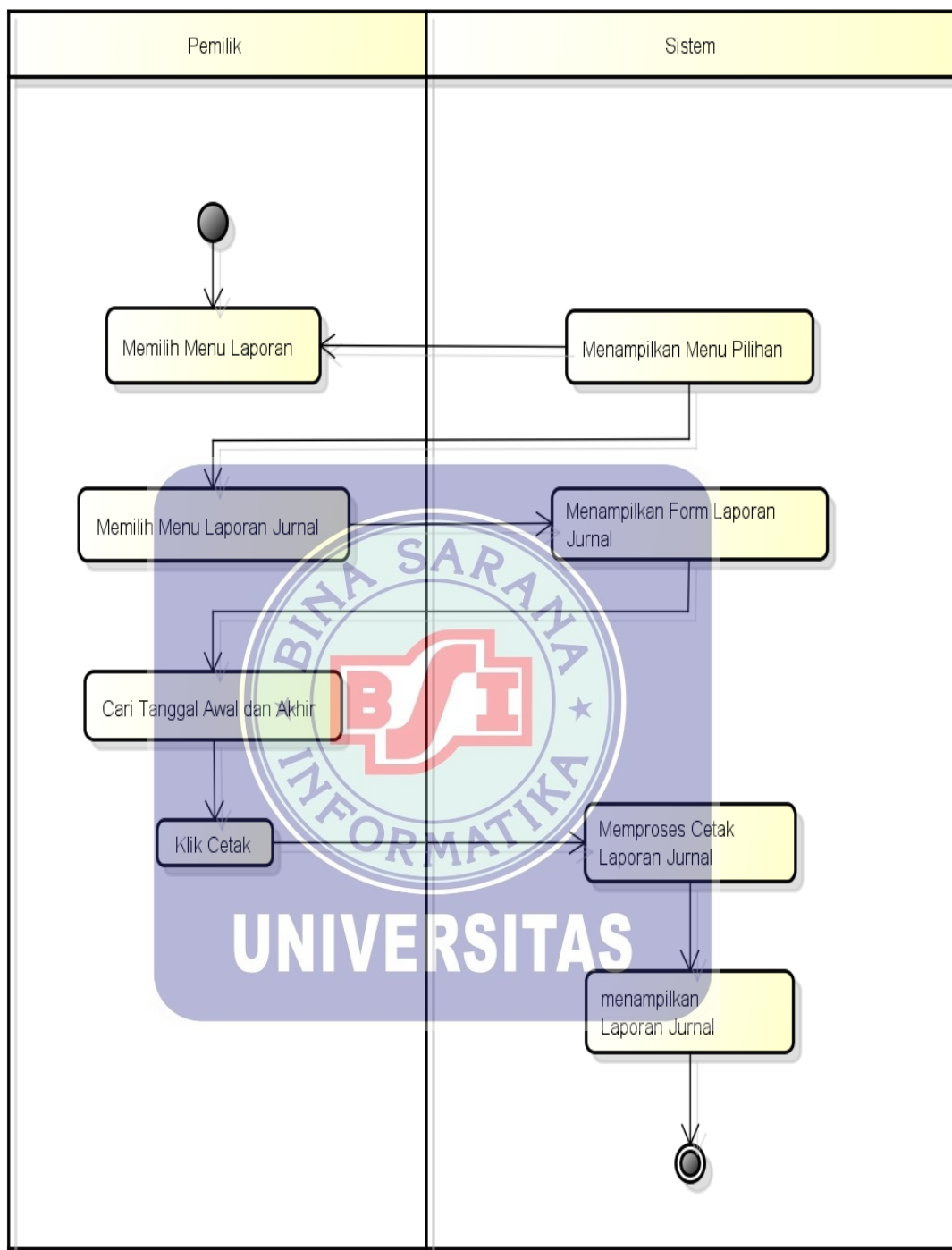


powered by Astah

Gambar III.16.
Activity Diagram Administrasi mengelola Laporan Pendapatan

B. Pemilik**B.1. Activity Diagram Pemilik melakukan Login****Gambar III.17.****Activity Diagram Pemilik Melakukan Login**

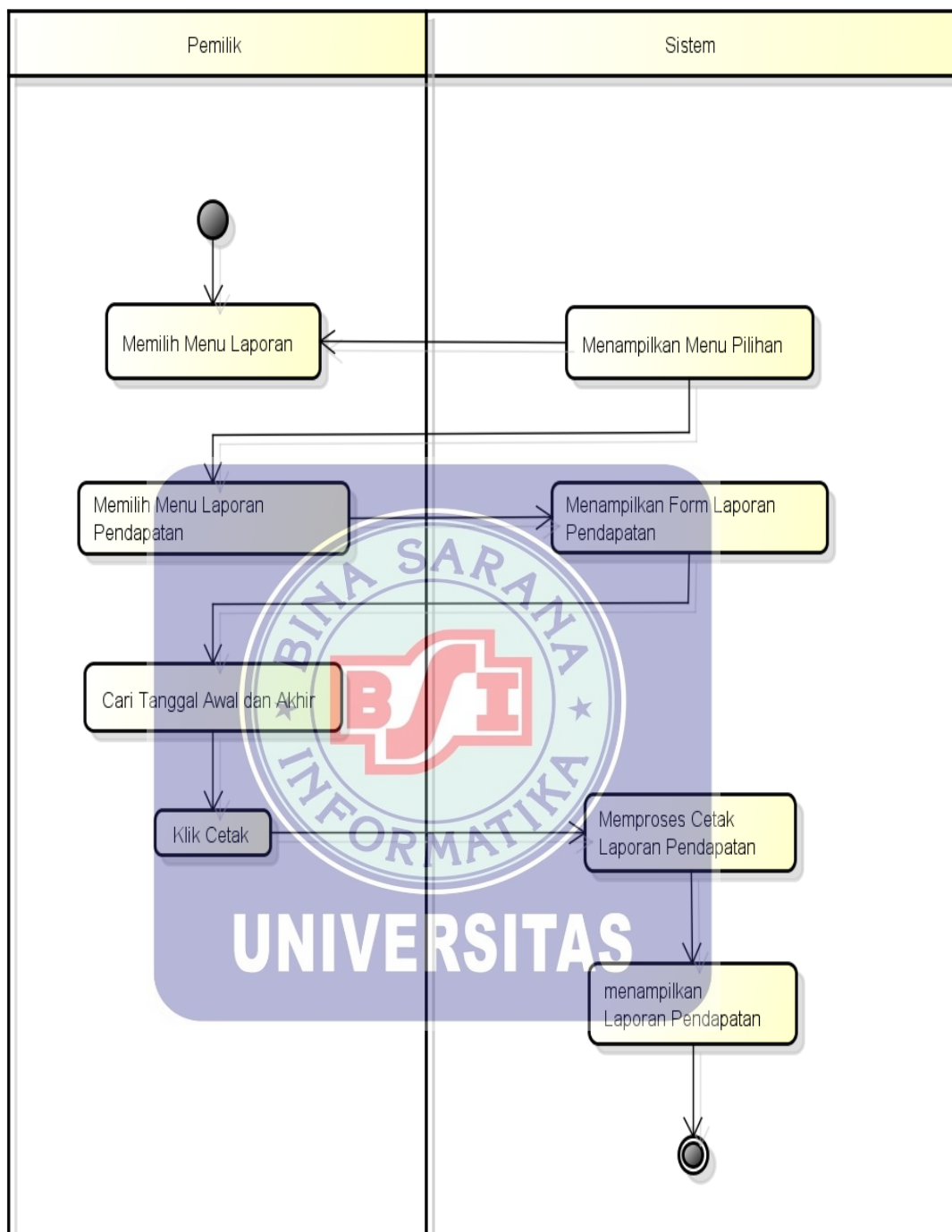
B.2. Activity Diagram Pemilik mengakses Laporan Jurnal



powered by Astah

Gambar III.18.
Activity Diagram Pemilik Mengakses Laporan Jurnal

B.3 Activity Diagram Pemilik mengakses Laporan Pendapatan

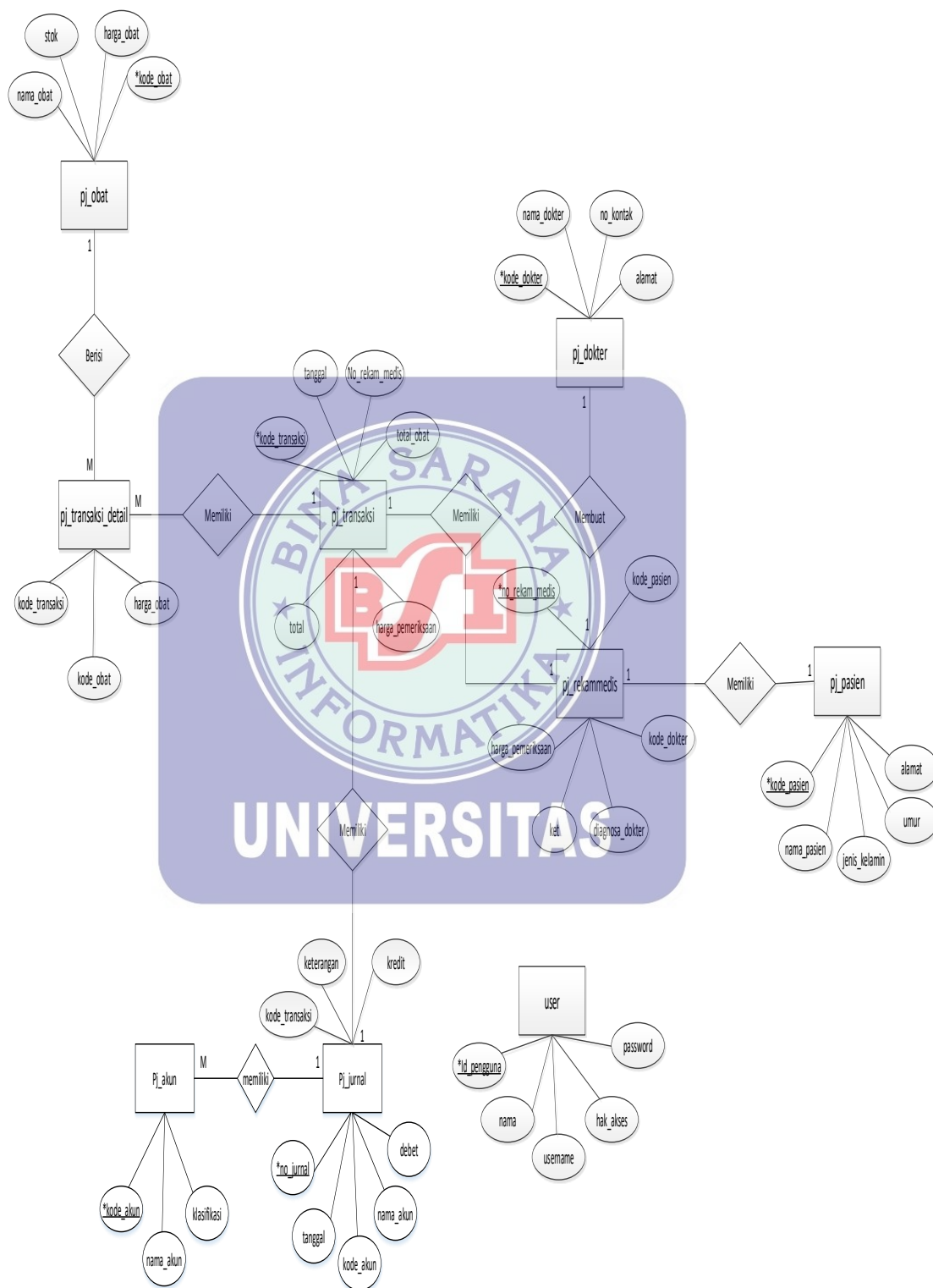


powered by Astah

Gambar III.19.
Activity Diagram Pemilik Mengakses Laporan Pendapatan

3.4. Desain

3.4.1. Entity Relationship Diagram (ERD)



Gambar III.20.
Entity Relationship Diagram (ERD) Pelayanan Rawat Jalan

3.4.2. Logical Record Structure (LRS)



Gambar III.21.
Logical Record Structure (LRS) Pelayanan Rawat Jalan

3.4.3. Spesifikasi File

Program ini menggunakan *Database* dengan nama klinik dan didalamnya terdapat tabel-tabel sebagai entitas. Tabel-tabel tersebut sebagai berikut:

1. Spesifikasi *File* Tabel Login

Nama *Database* : klinik

Nama *File* : *user*

Tipe *File* : Master

Akses *File* : *Random*

Record Size : 70 Karakter

Filed Key : id_pengguna

Software : MySQL

Tabel III.3.
Spesifikasi *File* Tabel User

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Id pengguna	id_pengguna	<i>varchar</i>	5	<i>Primary Key</i>
2	Nama	Nama	<i>varchar</i>	20	
3	<i>Username</i>	<i>username</i>	<i>varchar</i>	10	
4	<i>Password</i>	<i>password</i>	<i>varchar</i>	15	
5	Hak akses	hak_akses	<i>varchar</i>	20	

2. Spesifikasi *File* Tabel Data Akun

Nama *Database* : klinik

Nama *File* : pj_akun

Tipe *File* : Master

Akses *File* : *Random*

Record Size : 35 Karakter

Filed Key : kode_akun

Software : MySQL

Tabel III.4.
Spesifikasi File Tabel Data Akun

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Kode Akun	kode_akun	<i>varchar</i>	5	<i>Primary Key</i>
2	Nama Akun	nama_akun	<i>varchar</i>	20	
3	Klasifikasi	klasifikasi	<i>varchar</i>	10	

3. Spesifikasi File Tabel Data Dokter

Nama Database : klinik

Nama File : pj_dokter

Tipe File : Master

Akses File : *Random*

Record Size : 34 Karakter

Filed Key : kode_dokter

Software : MySQL

Tabel III.5.
Spesifikasi File Tabel Data Dokter

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Kode Dokter	kode_dokter	<i>varchar</i>	7	<i>Primary Key</i>
2	Nama Dokter	nama_dokter	<i>varchar</i>	15	
3	No Kontak	no_kontak	<i>varchar</i>	12	
4	Alamat	alamat	<i>Text</i>		

4. Spesifikasi File Tabel Data Obat

Nama Database : klinik

Nama File : pj_obat

Tipe File : Master
 Akses File : Random
 Record Size : 57 Karakter
 Filed Key : kode_obat
 Software : MySQL

Tabel III.6.
Spesifikasi File Tabel Data Obat

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Kode Obat	kode_obat	<i>varchar</i>	7	<i>Primary Key</i>
2	Nama Obat	nama_Obat	<i>varchar</i>	20	
3	Jumlah	stok	<i>int</i>	10	
4	Harga Obat	harga_obat	<i>int</i>	20	

5. Spesifikasi File Tabel Data Pasien

Nama Database : klinik
 Nama File : pj_pasien
 Tipe File : Master
 Akses File : Random
 Record Size : 42 Karakter
 Filed Key : kode_pasien
 Software : MySQL

Tabel III.7.
Spesifikasi File Tabel Data Pasien

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Kode Pasien	kode_pasien	<i>varchar</i>	7	<i>Primary Key</i>
2	Nama Pasien	nama_pasien	<i>varchar</i>	20	
3	Jenis Kelamin	jenis_kelamin	<i>varchar</i>	10	

4	Umur	umur	<i>Int</i>	5	
5	Alamat	alamat	<i>text</i>		

6. Spesifikasi *File* Tabel Data Rekam Medis

Nama *Database* : klinik
 Nama *File* : pj_rekammedis
 Tipe *File* : Master
 Akses *File* : *Random*
 Record Size : 81 Karakter
 Filed Key : no_rekam_medis
 Software : MySQL

Tabel III.8.
Spesifikasi *File* Tabel Data Rekam Medis

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	No Rekam Medis	no_rekam_medis	<i>varchar</i>	7	<i>Primary Key</i>
2	Kode Pasien	kode_pasien	<i>varchar</i>	7	<i>Foreign Key</i>
3	Kode Dokter	kode_dokter	<i>varchar</i>	7	<i>Foreign Key</i>
4	Diagnosa Dokter	diagnosa_dokter	<i>varchar</i>	20	
5	Keterangan	ket	<i>varchar</i>	20	
6	Harga Pemeriksaan	harga_pemeriksaan	<i>int</i>	20	

7. Spesifikasi *File* Transaksi Rawat Jalan

Nama *Database* : klinik
 Nama *File* : pj_transaksi
 Tipe *File* : Transaksi
 Akses *File* : *Random*
 Record Size : 73 Karakter

Filed Key : kode_transaksi

Software : MySQL

Tabel III.9.
Spesifikasi File Tabel Transaksi

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Kode Transaksi	kode_transaksi	<i>varchar</i>	6	<i>Primary Key</i>
2	Tanggal Transaksi	tanggal	<i>date</i>		
3	No Rekam Medis	no_rekam_medis	<i>vachar</i>	7	<i>Foreign Key</i>
4	Total Obat	total_obat	<i>int</i>	20	
5	Harga Pemeriksaan	harga_pemeriksaan	<i>int</i>	20	
6	Total	total	<i>int</i>	20	

8. Spesifikasi *File Transaksi Detail*

Nama *Database* : klinik

Nama *File* : pj_transaksi_detail

Tipe *File* : Transaksi

Akses *File* : *Random*

Record Size : 33 Karakter

Filed Key : -

Software : MySQL

Tabel III.10.
Spesifikasi File Tabel Transaksi Detail

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Kode transaksi	kode_transaksi	<i>Varchar</i>	6	<i>Foreign Key</i>
2	Kode obat	kode_obat	<i>Varchar</i>	7	<i>Foreign Key</i>
3	Harga obat	harga_obat	<i>Int</i>	20	

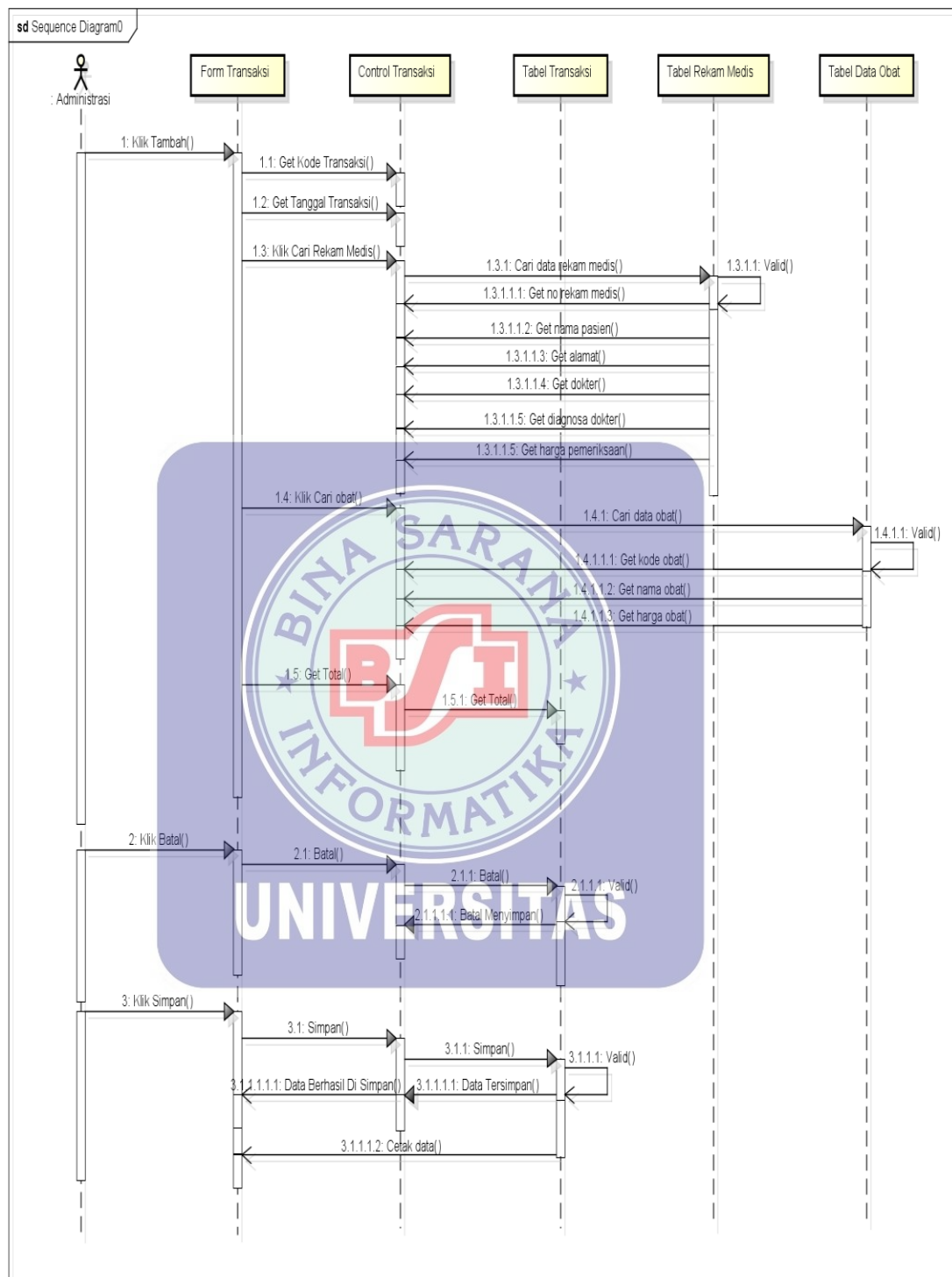
9. Spesifikasi *File* Jurnal

Nama *Database* : klinik
 Nama *File* : pj_jurnal
 Tipe *File* : Transaksi
 Akses *File* : *Random*
 Record Size : 98 Karakter
 Filed Key : no_jurnal
 Software : *MySQL*

Tabel III.11.
Spesifikasi *File* Tabel Jurnal

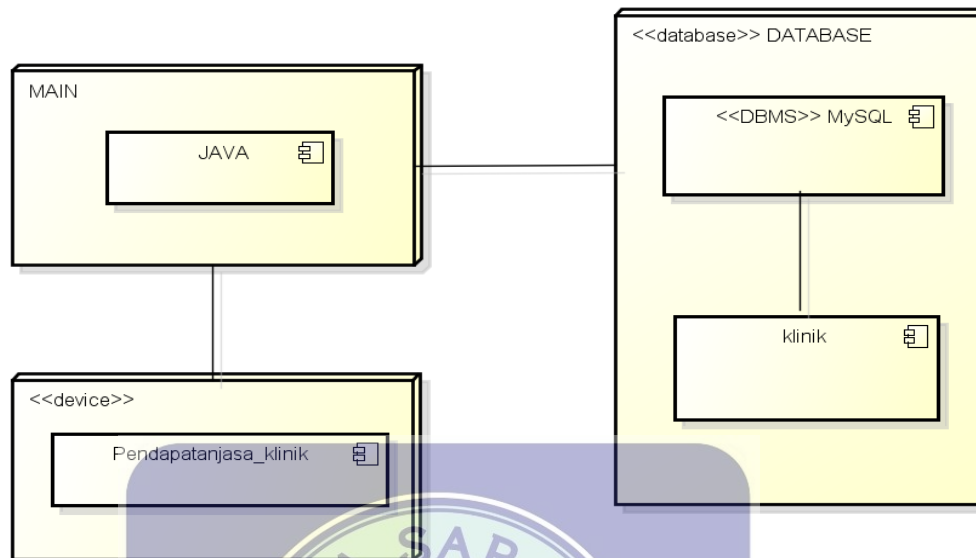
No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	No Jurnal	no_jurnal	<i>Int</i>	7	<i>Primary Key</i>
2	Tanggal	tanggal	<i>date</i>		
3	Kode Akun	kode_akun	<i>vachar</i>	5	<i>Foreign Key</i>
4	Nama Akun	nama_akun	<i>vachar</i>	20	
5	Debet	debet	<i>Int</i>	20	
6	Kredit	kredit	<i>Int</i>	20	
7	Keterangan	keterangan	<i>vachar</i>	20	
8	Kode Transaksi	kode_transaksi	<i>vachar</i>	6	<i>Foreign Key</i>

3.4.4. Sequence Diagram



Gambar III.22.
Sequence Diagram Transaksi Rawat Jalan

3.4.5. Depyoymment Diagram



Gambar III.23.
Deployment Diagram

3.4.6. User Interface

1. Administrasi

A.1. Admin Melakukan Login

Gambar III.24.
User Interface Administrasi Login

Tampilan Menu Utama Administrasi



Gambar III.25.
User Interface Menu Utama Administrasi

A.2. Administrasi Mengelola Data Akun

Kode Akun	Nama Akun	klasifikasi
1-1001	Kas	Harta
4-1001	Pendapatan	Pendapatan

Gambar III.26.
User Interface Data Akun

A.3. Administrasi mengelola Data Dokter



Klinik Vidya Medika

DATA DOKTER

Kode Dokter No Kontak

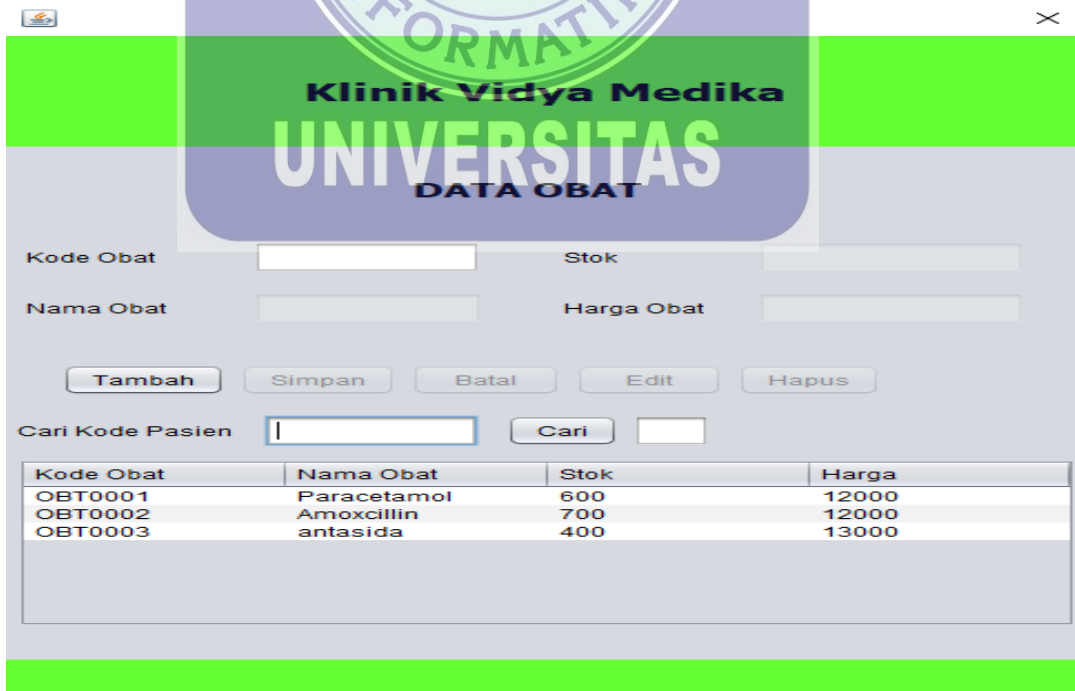
Nama Dokter Alamat

Cari Kode Dokter

Kode Dokter	Nama Dokter	No Kontak	Alamat
DKR0001	Dr. Alpa	08989426674	Karawang
DKR0002	Dr. Reza	08989426674	Cikampek

Gambar III.27.
User Interface Data Dokter

A.4 Administrasi mengelola Data Obat



Klinik Vidya Medika

DATA OBAT

Kode Obat Stok

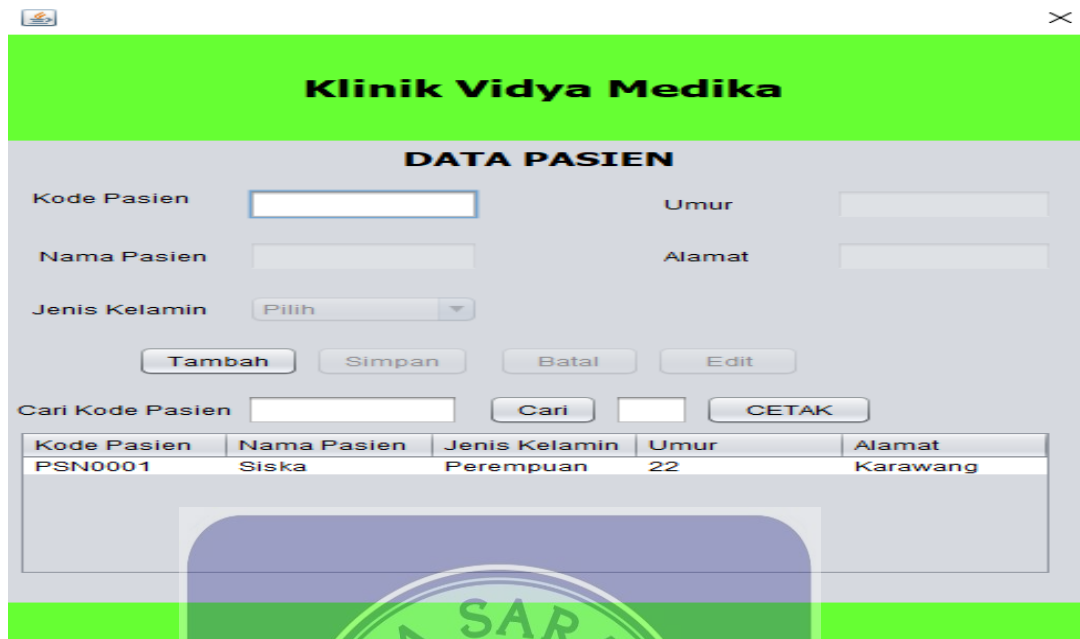
Nama Obat Harga Obat

Cari Kode Pasien

Kode Obat	Nama Obat	Stok	Harga
OBT0001	Paracetamol	600	12000
OBT0002	Amoxcillin	700	12000
OBT0003	antasida	400	13000

Gambar III.28.
User Interface Data Obat

A.5. Administrasi mengelola Data Pasien



Klinik Vidya Medika

DATA PASIEN

Kode Pasien Umur

Nama Pasien Alamat

Jenis Kelamin

Cari Kode Pasien

Kode Pasien	Nama Pasien	Jenis Kelamin	Umur	Alamat
PSN0001	Siska	Perempuan	22	Karawang

Gambar III.29.
User Interface Data Pasien

A.6. Adminstrasi mengelola Data Rekam Medis



Klinik Vidya Medika

DATA REKAM MEDIS

No. Rekam Medis Dokter

Kode Pasien Diagnosa Dokter

Nama Pasien Keterangan

Alamat Harga Pemeriksaan

Kode Dokter

Cari Rekam Medis

No Reka...	Kode Pa...	Nama P...	Alamat	Kode Do...	Nama D...	Diagnos...	Ket	Harga P...
RKM0001	PSN0001	siska	karawang	DKR0001	Dr. Alpa	panas	rawat jalan	75000

Gambar III.30.
User Interface Data Rekam Medis

A.7. Administrasi Mengelola Data Transaksi Rawat Jalan

TRANSAKSI RAWAT JALAN

Kode Transaksi: Diagnosa Dokter:

Tanggal Transaksi: Kode Obat:

No. Rekam Medis: Nama Obat:

Nama Pasien: Harga Obat:

Alamat: Harga Pemeriksaan:

Dokter: Total:

Cari Kode Transaksi:

Kode T...	Tanggal	No Rek...	Nama ...	Alamat	Harga ...	Total O...	Total	Kode O...	Nama ...	Harga ...
TRS001	2019-0...	RKM00...	siska	Karawa...	75000	23000	98000	OBT00...	Paracet...	12000
								OBT00...	Amoxcil...	11000

Gambar III.31.
User Interface Transaksi Rawat Jalan

A.8. Administrasi mengelola Data Jurnal

DATA JURNAL

No Jurnal:

No Jurnal	Tanggal	Kode Akun	Nama Akun	debit	kredit	Keterangan	ID Transaksi
39	2019-09-18	1-100	Kas	98000	0	Penerimaan	TRS001
40	2019-09-18	4-100	Pendapatan	0	98000	Penerimaan	TRS001

Gambar III.32.
User Interface Jurnal

A.9 Administrasi Mengelola Data Laporan Jurnal



Gambar III.33.
User Interface Laporan Jurnal

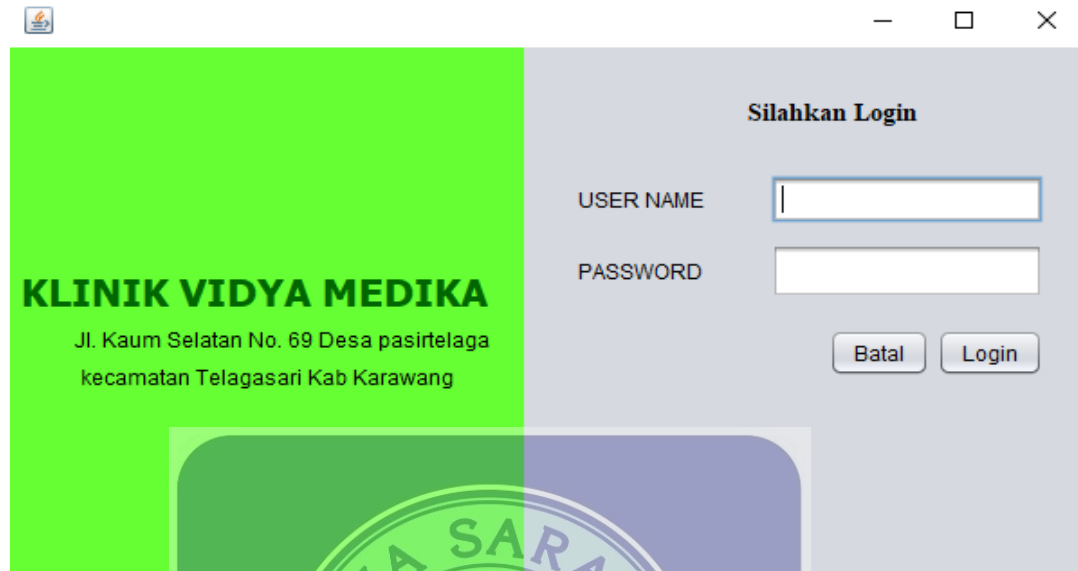
A.10 Administrasi Mengelola Data Laporan Pendapatan



Gambar III.34.
User Interface Laporan Pendapatan

2. Pemilik

B.1 Pemilik melakukan Login



KLINIK VIDYA MEDIKA
Jl. Kaum Selatan No. 69 Desa pasirtelaga
kecamatan Telagasari Kab Karawang

Silahkan Login

USER NAME

PASSWORD

Batal Login

Gambar III.35.
User Interface Pemilik Login

Tampilan Menu Utama Pemilik



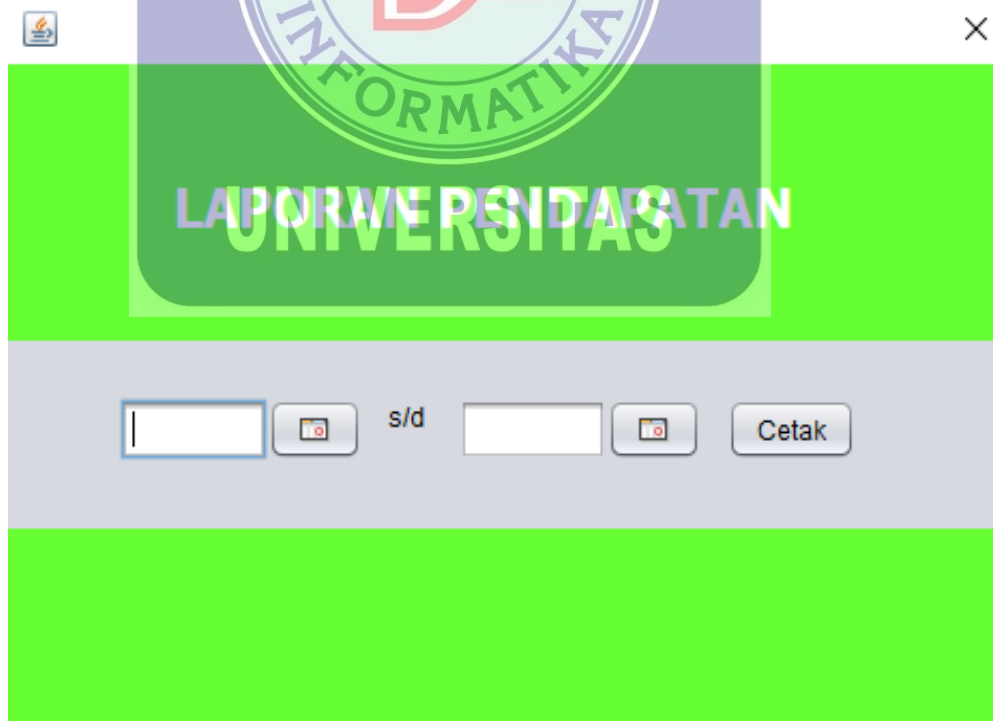
Gambar III.36.
User Interface Menu Utama Pemilik

B.2. Pemilik berhak mengakses laporan Jurnal



Gambar III.37.
User Interface Laporan Jurnal

B.3 Pemilik berhak mengakses Laporan Pendapatan



Gambar III.38.
User Interface Laporan Pendapatan

3.5. Implementasi

3.5.1. Code Generation

1. Transaksi Rawat Jalan

```
import pendapatanjasa_melinda_ta.Koneksi2;
import java.awt.Toolkit;
import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperCompileManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.design.JasperDesign;
import net.sf.jasperreports.engine.xml.JRXmlLoader;
import net.sf.jasperreports.view.JasperViewer;
import java.io.File;
import java.sql.Connection;
import java.text.DateFormat;
import java.util.Calendar;
import java.util.HashMap;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JDialog;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

Koneksi2 conn = new Koneksi2();
String kode, tanggal, penerima, keterangan, ubah;
int jml;
int harga = 0;
Date txttgl;
ResultSet resultSet;
java.sql.Statement statement;
DefaultTableModel tb;
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
int loop = 1
_melinda_ta.Koneksi2 konek = new pendapatanjasa_melinda_ta.Koneksi2();
public FormRawatJalan(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
    Calendar cal = Calendar.getInstance();
    ttanggal.setText(dateFormat.format(cal.getTime()));
    tabelmodel();
    akun.hide();
    ket.hide();
}
```

```

    kredit.hide();
    nonaktif();
    setkode();
    pen.hide();
    no_jurnal.hide();
    akun1.hide();
    ket1.hide();
}
private void nonaktif() {
    simpan.setEnabled(false);
    batal.setEnabled(false);
    cetak.setEnabled(false);
}
public void tabelmodel() {
    tb = new DefaultTableModel();
    tb.addColumn("Kode Transaksi");
    tb.addColumn("Tanggal");
    tb.addColumn("No Rekam Medis");
    tb.addColumn("Nama Pasien");
    tb.addColumn("Alamat");
    tb.addColumn("Harga Periksa");
    tb.addColumn("Total Obat");
    tb.addColumn("Total");
    tb.addColumn("Kode Obat");
    tb.addColumn("Nama Obat");
    tb.addColumn("Harga Obat");
    jTable1.setModel(tb);
    String Id = "";
    try {
        ResultSet res = conn.ambilData("select *from pj_transaksi INNER JOIN "
            + "pj_transaksi_detail ON "
            + "pj_transaksi.kode_transaksi=pj_transaksi_detail.kode_transaksi INNER JOIN "
            + "pj_obat ON pj_obat.kode_obat=pj_transaksi_detail.kode_obat "
            + "INNER JOIN "
            + "pj_rekammedis ON "
            + "pj_rekammedis.no_rekam_medis=pj_transaksi.no_rekam_medis INNER JOIN "
            + "pj_pasien ON pj_pasien.kode_pasien= pj_rekammedis.kode_pasien "
            + "ORDER BY pj_transaksi.kode_transaksi");
        while (res.next()) {
            if (!Id.equals(res.getString("kode_transaksi"))) {
                tb.addRow(new Object[] {
                    res.getString("kode_transaksi"),
                    res.getString("tanggal"),
                    res.getString("no_rekam_medis"),
                    res.getString("nama_pasien"),
                    res.getString("alamat"),
                    res.getString("harga_pemeriksaan"),
                    res.getString("total_obat"),
                    res.getString("total"),
                });
            }
        }
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, ex.getMessage());
    }
}

```

```

        res.getString("kode_obat"),
        res.getString("nama_obat"),
        res.getString("harga_obat")
    });
    } else {
        tb.addRow(new Object[]{
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            res.getString("kode_obat"),
            res.getString("nama_obat"),
            res.getString("harga_obat")
        });
    }
    Id = res.getString("kode_transaksi");
}
 jTable1.setModel(tb);
} catch (SQLException ex) {
    JOptionPane.showMessageDialog(this, ex);
}
}

public int simpan() {
    String sql = "";
    System.out.println(harga);
    if (loop == 1) {
        sql = "insert into pj_transaksi "
            + "values"
            + "(" + idt.getText() + ", "
            + "" + ttanggal.getText() + ", "
            + "" + norekam.getText() + ", "
            + "" + hobat.getText() + ", "
            + "" + hperiksa.getText() + ", "
            + "" + total.getText() + " " + ")";
        conn.simpanData(sql);

        hapus_detail(idt.getText(), obat.getText());

        sql = "insert into pj_transaksi_detail "
            + "values"
            + "(" + idt.getText() + ", "
            + "" + obat.getText() + ", "
            + "" + String.valueOf(harga) + ")";
        conn.simpanData(sql);
    }
}

```

```

    } else {
        hapus_detail(idt.getText(), obat.getText());
        sql = "insert into pj_transaksi_detail "
            + "values"
            + "(" + idt.getText() + ", "
            + "" + obat.getText() + ", "
            + "" + String.valueOf(harga) + ")";
        conn.simpanData(sql);

        sql = "UPDATE pj_transaksi set total_obat=" + hobat.getText() + ", total="
+ Integer.parseInt(total.getText()) + " where kode_transaksi=" + idt.getText() + """;
        conn.simpanData(sql);
    }
    return 0;
}

public int simpanpendapatan() {
    String sql = ("INSERT INTO pj_jurnal (tanggal, kode_akun, nama_akun, debet,
kredit, keterangan, kode_transaksi) VALUES (" + ttanggal.getText() + ", " +
akun.getText() + ", " + ket.getText() + ", " + kredit.getText() + ", " +
total.getText() + ", " + pen.getText() + ", " + idt.getText() + ")");
    conn.simpanData(sql);
    return 0;
}

public int simpankas() {
    String sql = ("INSERT INTO pj_jurnal (tanggal, kode_akun, nama_akun, debet,
kredit, keterangan, kode_transaksi) VALUES (" + ttanggal.getText() + ", " +
akun1.getText() + ", " + ket1.getText() + ", " + total.getText() + ", " +
kredit.getText() + ", " + pen.getText() + ", " + idt.getText() + ")");
    conn.simpanData(sql);
    return 0;
}

public int hapus(String kd) {
    String sql = "delete from transaksi where id_transaksi = " + kd + """;
    conn.simpanData(sql);
    return 0;
}

public int hapus_detail(String kd, String obat) {
    String sql = "delete from pj_transaksi_detail where kode_transaksi = " + kd + "
AND kode_obat=" + obat + """;
    conn.simpanData(sql);
    return 0;
}

void bersih() {
    norekam.setText("");
    pasien.setText("");
    alamat.setText("");
    dokter.setText("");
    diagnosa.setText("");
}

```

```

        obat.setText("");
        nobat.setText("");
        hobat.setText("");
        hperiksa.setText("");
        total.setText("");
    }
    int totalobat() {
        int total = 0;
        try {
            ResultSet res = conn.ambilData("select sum(harga_obat) as total from
            pj_transaksi_detail where kode_transaksi=" + idt.getText() + "");
            if (res.next()) {
                total = res.getInt("total");
            }
        } catch (Exception ex) {

        }
        return total;
    }
    void setkode() {
        try {
            ResultSet res = conn.ambilData("select * from pj_transaksi order by
            kode_transaksi desc");
            if (res.next()) {
                String nofak = res.getString("kode_transaksi").substring(3);
                String AN = "" + (Integer.parseInt(nofak) + 1);
                String Nol = "";
                if (AN.length() == 1) {
                    Nol = "00";
                } else if (AN.length() == 2) {
                    Nol = "0";
                } else if (AN.length() == 3) {
                    Nol = "";
                }
                idt.setText("TRS" + Nol + AN);
            } else {
                idt.setText("TRS001");
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }
    void setkodee() {
        try {
            ResultSet res = conn.ambilData("select * from pj_jurnal order by no_jurnal
            desc");
            if (res.next()) {
                String nofak = res.getString("no_jurnal").substring(3);
                String AN = "" + (Integer.parseInt(nofak) + 1);
            }
        }
    }

```

```

        String Nol = "";
        if (AN.length() == 1) {
            Nol = "00";
        } else if (AN.length() == 2) {
            Nol = "0";
        } else if (AN.length() == 3) {
            Nol = "";
        }
        no_jurnal.setText("JRN" + Nol + AN);
    } else {
        no_jurnal.setText("JRN001");
    }
}

} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e);
}
}

```

```

jPanel1 = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();
jPanel2 = new javax.swing.JPanel();
txtcari = new javax.swing.JTextField();
jButton6 = new javax.swing.JButton();
batal = new javax.swing.JButton();
pasien = new javax.swing.JTextField();
dokter = new javax.swing.JTextField();
alamat = new javax.swing.JTextField();
nobat = new javax.swing.JTextField();
hobat = new javax.swing.JTextField();
jLabel13 = new javax.swing.JLabel();
jScrollPane1 = new javax.swing.JScrollPane();
jTable1 = new javax.swing.JTable();
hperiksa = new javax.swing.JTextField();
jLabel2 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
diagnosa = new javax.swing.JTextField();
jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
simpan = new javax.swing.JButton();
jLabel7 = new javax.swing.JLabel();
jButton3 = new javax.swing.JButton();
jLabel8 = new javax.swing.JLabel();
jLabel15 = new javax.swing.JLabel();
jLabel9 = new javax.swing.JLabel();
total = new javax.swing.JTextField();
jLabel16 = new javax.swing.JLabel();
jLabel11 = new javax.swing.JLabel();
idt = new javax.swing.JTextField();

```

```

norekam = new javax.swing.JTextField();
obat = new javax.swing.JTextField();
ttanggal = new javax.swing.JTextField();
akun = new javax.swing.JTextField();
no_jurnal = new javax.swing.JTextField();
ket = new javax.swing.JTextField();
kredit = new javax.swing.JTextField();
jButton1 = new javax.swing.JButton();
jButton4 = new javax.swing.JButton();
cetak = new javax.swing.JButton();
akun1 = new javax.swing.JTextField();
ket1 = new javax.swing.JTextField();
pen = new javax.swing.JTextField();
jLabel10 = new javax.swing.JLabel();
setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
jPanel1.setBackground(new java.awt.Color(102, 255, 51));
jLabel1.setFont(new java.awt.Font("Tahoma", 1, 22)); // NOI18N
jLabel1.setText("TRANSAKSI RAWAT JALAN");
javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(198, 198, 198)
        .addComponent(jLabel1)
        .addGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    );
jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(19, 19, 19)
        .addComponent(jLabel1)
        .addGap(19, 19, 19)
    );
jPanel2.setBackground(new java.awt.Color(102, 255, 51));
javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGap(0, 0, Short.MAX_VALUE)
);

```

```

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
    .addGap(0, 96, Short.MAX_VALUE)
);
jButton6.setText("Cari");
jButton6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton6ActionPerformed(evt);
    }
});
batal.setText("Batal");
batal.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        batalActionPerformed(evt);
    }
});
pasien.setEditable(false);
pasien.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        pasienActionPerformed(evt);
    }
});
dokter.setEditable(false);
alamat.setEditable(false);
nobat.setEditable(false);
hobat.setEditable(false);
jLabel13.setText("Harga Pemeriksaan ");
jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jTable1.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jTable1MouseClicked(evt);
    }
});
jScrollPane1.setViewportViewView(jTable1);
hperiksa.setEditable(false);
jLabel2.setText("Kode Transaksi");

```

```

jLabel14.setText("Diagnosa Dokter");
diagnosa.setEditable(false);
jLabel3.setText("Tanggal Transaksi");
jLabel4.setText("No. Rekam Medis");
jLabel5.setText("Nama Pasien");
simpan.setText("Simpan");
simpan.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        simpanActionPerformed(evt);
    }
});
jLabel7.setText("Dokter");
jButton3.setText("Tambah");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});
jLabel8.setText("Alamat");
jLabel15.setText("Total");
jLabel9.setText("Kode Obat");
total.setEditable(false);
jLabel16.setText("Cari Kode Transaksi");
jLabel11.setText("Harga Obat");
idt.setEditable(false);
akun.setEditable(false);
akun.setText("4-100");
no_jurnal.setEditable(false);
ket.setEditable(false);
ket.setText("Pendapatan");
ket.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ketActionPerformed(evt);
    }
});
kredit.setEditable(false);
kredit.setText("0");
jButton1.setText("Cari");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
jButton4.setText("Cari");
jButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton4ActionPerformed(evt);
    }
});

```



```

        .addComponent(pasien,
javax.swing.GroupLayout.PREFERRED_SIZE, 102,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(layout.createSequentialGroup())
        .addComponent(norekam,
javax.swing.GroupLayout.PREFERRED_SIZE, 102,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jButton1))
        .addComponent(alamat,
javax.swing.GroupLayout.PREFERRED_SIZE, 102,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(dokter,
javax.swing.GroupLayout.PREFERRED_SIZE, 102,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(56, 56, 56))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        NG)
        .addComponent(idt,
javax.swing.GroupLayout.PREFERRED_SIZE, 102,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(ttanggal,
javax.swing.GroupLayout.PREFERRED_SIZE, 102,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(113, 113, 113)))

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        NG)
        .addComponent(jLabel9)
        .addComponent(jLabel14)
        .addComponent(jLabel11)
        .addComponent(jLabel13)
        .addComponent(jLabel15,
javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel10))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        NG)

```

```

        .addComponent(total,
javax.swing.GroupLayout.PREFERRED_SIZE, 102,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(layout.createSequentialGroup())

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
        .addComponent(nobat,
javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(diagnosa,
javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(obat,
javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(hobat,
javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(hperiksa,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 102,
javax.swing.GroupLayout.PREFERRED_SIZE))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jButton4)))
        .addContainerGap(39,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())
        .addGap(20, 20, 20)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())
        .addComponent(akun1,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(ket1,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(pen,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addComponent(jLabel16)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtcari,
javax.swing.GroupLayout.PREFERRED_SIZE, 114,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jButton6)
            .addGap(18, 18, 18)
            .addComponent(akun,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(ket,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(no_jurnal,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(kredit,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(layout.createSequentialGroup()
            .addComponent(jButton3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(simpan)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(batal,
javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(cetak,
javax.swing.GroupLayout.PREFERRED_SIZE, 77,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 361,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addComponent(jScrollPane1))
    .addGap(14, 14, 14)))
    .addComponent(jPanel2, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel2)
    .addComponent(idt, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel14)
    .addComponent(diagnosa,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel3)
    .addComponent(jLabel9)
    .addComponent(obat, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(ttanggal,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jButton4))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jLabel4)
        .addGap(23, 23, 23)

```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jLabel5)
    .addComponent(pasien,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(17, 17, 17)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jLabel8)
    .addComponent(alamat,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jLabel7)
    .addComponent(dokter,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addGroup(layout.createSequentialGroup())
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(norekam,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jButton1)
    .addComponent(nobat,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel10))
    .addGap(18, 18, 18)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(hobat,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel11))
```

```

.addGap(20, 20, 20)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(hperiksa,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel13))
.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel15)
    .addComponent(total,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(8, 8, 8)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jButton3)
    .addComponent(simpan)
    .addComponent(batal)
    .addComponent(cetak))
.addGap(18, 18, 18))
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
    .addComponent(pen, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(akun1,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(ket1,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

        .addGap(3, 3, 3)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel16)
    .addComponent(txtcari, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jButton6)
    .addComponent(akun, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(no_jurnal,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(ket, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(kredit, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 121,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap()
);

setSize(new java.awt.Dimension(710, 662));
setLocationRelativeTo(null);
} // </editor-fold>

private void pasienActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void simpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (total.getText().equals("") || idt.getText().equals("")) {
        JOptionPane.showMessageDialog(this, "Semuah Data Harus Di Isi");
    } else {
        simpan();
    }
}

```

```

        if (JOptionPane.showConfirmDialog(rootPane, "Tambah rincian obat?",
"Konfirmasi", JOptionPane.YES_NO_OPTION) == JOptionPane.NO_OPTION) {
            simpankas();
            simpanpendapatan();
            nonaktif();
            tabelmodel();
            JOptionPane.showMessageDialog(null, "Berhasil menyimpan data
transaksi");

            try {
                HashMap hash = new HashMap();
                hash.put("kode_transaksi", idt.getText());

                File file = new File("src/Laporan/bukti_trans.jrxml");
                JasperDesign jasperDesign = JRXmlLoader.load(file);
                JasperReport jasperReport =
JasperCompileManager.compileReport(jasperDesign);
                JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport,
hash, konek.openkoneksi());

                JDialog view = new JDialog(this);
                view.setSize(Toolkit.getDefaultToolkit().getScreenSize());
                view.setTitle("Bukti Transaksi");
                view.setLocation(0, 0);
                JasperViewer jv = new JasperViewer(jasperPrint, false);
                view.setContentPane(jv.getContentPane());
                view.setVisible(true);

            } catch (ClassNotFoundException | JRException e) {
                JOptionPane.showMessageDialog(null, "Error " + e);
            }

            bersih();

        } else {
            loop++;
            obat.setText("");
            nobat.setText("");
        }
    }

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    simpan.setEnabled(true);
    batal.setEnabled(true);
    cetak.setEnabled(true);
}

```

```

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int row = jTable1.getSelectedRow();
    String row_id = (jTable1.getModel().getValueAt(row, 0).toString());
    idt.setText(row_id);
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    harga = 0;
    obat.setText(new FormCariObat(null, true).getKodeObat());
    try {

        ResultSet res = conn.ambilData("select * from pj_obat where " +
"kode_obat=" + obat.getText()
        + "");
        if (res.next()) {
            obat.setText(res.getString("kode_obat"));
            nobat.setText(res.getString("nama_obat"));
            harga = res.getInt("harga_obat");
        } else {
            JOptionPane.showMessageDialog(null, "id obat tidak ditemukan.");
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(rootPane, e);
    }

    System.out.println(totalobat());
    int nilai1;
    int nilai2;
    int nilai3;
    String Hasil;
    nilai1 = (totalobat() == 0 ? harga : totalobat() + harga);
    nilai2 = Integer.valueOf(hperiksa.getText());
    nilai3 = nilai1 + nilai2;
    Hasil = String.valueOf(nilai3);
    hobat.setText(String.valueOf(nilai1));
    total.setText(Hasil);
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        norekam.setText(new FormCariRekamMedis(null, true).getNoRM());
        ResultSet res = conn.ambilData("select * from pj_rekammedis inner join "

```

```

        + "pj_pasien on pj_pasien.kode_pasien=pj_rekammedis.kode_pasien
inner join "
        + "pj_dokter on pj_dokter.kode_dokter=pj_rekammedis.kode_dokter
where no_rekam_medis=" + norekam.getText()
        + """);
    if (res.next()) {
        norekam.setText(res.getString("no_rekam_medis"));
        pasien.setText(res.getString("nama_pasien"));
        alamat.setText(res.getString("alamat"));
        dokter.setText(res.getString("nama_dokter"));
        diagnosa.setText(res.getString("diagnosa_dokter"));
        hperiksa.setText(res.getString("harga_pemeriksaan"));
    } else {
        JOptionPane.showMessageDialog(null, "No Rekam Medis tidak
ditemukan.");
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(rootPane, e);
}
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    cetak.setEnabled(true);
    DefaultTableModel tb = new DefaultTableModel();
    tb.addColumn("Kode Transaksi");
    tb.addColumn("Tanggal");
    tb.addColumn("No Rekam Medis");
    tb.addColumn("Nama Pasien");
    tb.addColumn("Alamat");
    tb.addColumn("Harga Periksa");
    tb.addColumn("Total Obat");
    tb.addColumn("Total");
    tb.addColumn("Kode Obat");
    tb.addColumn("Nama Obat");
    tb.addColumn("Harga Obat");
    try {
        Connection conn = konek.openkoneksi();
        java.sql.Statement stm = conn.createStatement();

        java.sql.ResultSet sql = stm.executeQuery("select *from pj_transaksi INNER
JOIN "
        + "pj_transaksi_detail ON
pj_transaksi.kode_transaksi=pj_transaksi_detail.kode_transaksi INNER JOIN "
        + "pj_obat ON pj_obat.kode_obat=pj_transaksi_detail.kode_obat
INNER JOIN "
        + "pj_rekammedis ON
pj_rekammedis.no_rekam_medis=pj_transaksi.no_rekam_medis INNER JOIN "

```

```

        + "pj_pasien ON pj_pasien.kode_pasien= pj_rekammedis.kode_pasien
WHERE pj_transaksi.kode_transaksi like '%" + txtcari.getText() + "%' OR
pj_rekammedis.no_rekam_medis like '%" + txtcari.getText() + "%' ORDER BY
pj_transaksi.kode_transaksi");

```

```

    String Id = "";

```

```

    while (sql.next()) {

```

```

        if (!Id.equals(sql.getString("kode_transaksi"))) {

```

```

            tb.addRow(new Object[]{

```

```

                sql.getString("kode_transaksi"),

```

```

                sql.getString("tanggal"),

```

```

                sql.getString("no_rekam_medis"),

```

```

                sql.getString("nama_pasien"),

```

```

                sql.getString("alamat"),

```

```

                sql.getString("harga_pemeriksaan"),

```

```

                sql.getString("total_obat"),

```

```

                sql.getString("total"),

```

```

                sql.getString("kode_obat"),

```

```

                sql.getString("nama_obat"),

```

```

                sql.getString("harga_obat")

```

```

            });

```

```

        } else {

```

```

            tb.addRow(new Object[]{

```

```

                "",

```

```

                "",

```

```

                "",

```

```

                "",

```

```

                "",

```

```

                "",

```

```

                "",

```

```

                "",

```

```

                sql.getString("kode_obat"),

```

```

                sql.getString("nama_obat"),

```

```

                sql.getString("harga_obat")

```

```

            });

```

```

        }

```

```

        Id = sql.getString("kode_transaksi");
    }

```

```

    jTable1.setModel(tb);

```

```

    } catch (Exception e) {

```

```

    }

```

```

}

```

```

private void batalActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    // TODO add your handling code here:

```

```

    tabelmodel();

```

```

    bersih();

```

```

    nonaktif();

```

```

    }

    private void cetakActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try {
            HashMap hash = new HashMap();
            hash.put("kode_transaksi", txtcari.getText());

            File file = new File("src/Laporan/bukti_trans.jrxml");
            JasperDesign jasperDesign = JRXmlLoader.load(file);
            JasperReport jasperReport =
JasperCompileManager.compileReport(jasperDesign);
            JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, hash,
koneksi.openkoneksi());

            JDialog view = new JDialog(this);
            view.setSize(Toolkit.getDefaultToolkit().getScreenSize());
            view.setTitle("Bukti Transaksi");
            view.setLocation(0, 0);
            JasperViewer jv = new JasperViewer(jasperPrint, false);
            view.setContentPane(jv.getContentPane());
            view.setVisible(true);

        } catch (ClassNotFoundException | JRException e) {
            JOptionPane.showMessageDialog(null, "Error " + e);
        }
        bersih();
    }

    private void ketActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void ket1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(FormRawatJalan.class.getName()).log(java.util.l
ogging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

```

```

java.util.logging.Logger.getLogger(FormRawatJalan.class.getName()).log(java.util.l
ogging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

```

```

java.util.logging.Logger.getLogger(FormRawatJalan.class.getName()).log(java.util.l
ogging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```

java.util.logging.Logger.getLogger(FormRawatJalan.class.getName()).log(java.util.l
ogging.Level.SEVERE, null, ex);
    }
//</editor-fold>
//</editor-fold>

```

```

/* Create and display the dialog */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        FormRawatJalan dialog = new FormRawatJalan(new
javax.swing.JFrame(), true);
        dialog.addWindowListener(new java.awt.event.WindowAdapter() {
            @Override
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
        dialog.setVisible(true);
    }
});
}

```

```

// Variables declaration - do not modify
private javax.swing.JTextField akun;
private javax.swing.JTextField akun1;
private javax.swing.JTextField alamat;
private javax.swing.JButton batal;
private javax.swing.JButton cetak;
private javax.swing.JTextField diagnosa;
private javax.swing.JTextField dokter;
private javax.swing.JTextField hobat;
private javax.swing.JTextField hperiksa;
private javax.swing.JTextField idt;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton6;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;

```

```
private javax.swing.JLabel jLabel13;  
private javax.swing.JLabel jLabel14;  
private javax.swing.JLabel jLabel15;  
private javax.swing.JLabel jLabel16;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabel7;  
private javax.swing.JLabel jLabel8;  
private javax.swing.JLabel jLabel9;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTable jTable1;  
private javax.swing.JTextField ket;  
private javax.swing.JTextField ket1;  
private javax.swing.JTextField kredit;  
private javax.swing.JTextField no_jurnal;  
private javax.swing.JTextField nobat;  
private javax.swing.JTextField norekam;  
private javax.swing.JTextField obat;  
private javax.swing.JTextField pasien;  
private javax.swing.JTextField pen;  
private javax.swing.JButton simpan;  
private javax.swing.JTextField total;  
private javax.swing.JTextField ttanggal;  
private javax.swing.JTextField txtcari;  
// End of variables declaration  
}
```



UNIVERSITAS

3.5.2. Blackbox Testing

1. Form Login Administrasi

Tabel III.12.
Hasil Pengujian Blackbox Testing Form Login

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Mengetikkan <i>Username</i> salah dan <i>password</i> salah kemudian klik tombol <i>login</i>	<i>Username:</i> Kty <i>Password:</i> 123456	Sistem akan menolak akses dan Menampilkan pesan “Maaf, <i>Username</i> atau <i>password</i> belum terdaftar“	Sesuai harapan	Valid
2	Mengetikkan <i>Username</i> di isi dan <i>password</i> tidak di isi kemudian klik tombol <i>login</i>	<i>Username:</i> admin <i>Password:</i> (kosong)	Sistem akan menolak akses dan menampilkan pesan “ <i>Password</i> Harus Di isi“	Sesuai harapan	Valid
3	<i>Username</i> tidak di isi dan <i>password</i> di isi kemudian klik tombol <i>login</i>	<i>Username:</i> (kosong) <i>Password:</i> 123	Sistem akan menolak akses dan menampilkan pesan “ <i>Username</i> Harus Di Isi“	Sesuai harapan	Valid
3	Mengetikkan <i>Username</i> dan <i>password</i> dengan data yang benar kemudian klik tombol <i>login</i>	<i>Username:</i> admin (benar) <i>Password:</i> 123 (benar)	Sistem akan menerima dan menampilkan pesan “Selamat Datang Anda Sebagai Administrasi”	Sesuai harapan	Valid

2. Form Data Pasien

Tabel III.13.

Hasil Pengujian Blackbox Testing Form Data Pasien

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Form tambah data pasien jika field isian data pasien dikosongkan kemudian klik simpan	Nama pasien: (kosong) Jenis Kelamin: (kosong) Umur: (kosong) Alamat: (kosong)	Sistem akan menolak dan menampilkan pesan "Semua Data Harus Di Isi"	Sesuai harapan	Valid
2	Mengisi semua field kemudian klik simpan	Nama pasien: (Siska) Jenis Kelamin: (Perempuan) Umur: (22) Alamat: (Karawang)	Sistem akan menerima dan menampilkan pesan "Berhasil Menyimpan Data Pasien"	Sesuai harapan	Valid
3	Pilih salah satu data pasien lalu klik edit	Ubah data pasien	Sistem akan menerima dan menampilkan pesan "Data Berhasil Diupdate"	Sesuai harapan	Valid

3. Form Transaksi Rawat Jalan

Tabel III.14.

Hasil Pengujian *Blackbox Testing* Form Transaksi Rawat Jalan

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Form tambah transaksi jika field isian dikosongkan kemudian klik simpan	No Rekam Medis: (kosong) Nama Pasien: (kosong) Alamat: (kosong) Dokter: (kosong) Diagnosa Dokter: (kosong) Kode obat: (kosong) Nama Obat: (kosong) Harga Obat: (Kosong) Harga Pemeriksaan: (Kosong) Total: (Kosong)	Sistem akan menolak dan menampilkan pesan “Semua Data Harus Diisi”	Sesuai harapan	Valid
2	Mengisi semua <i>text field</i>	Kode Transaksi: (TRS001) Tanggal Transaksi: (2019-09-18) No Rekam Medis: (RKM0001) Nama Pasien: (Siska) Alamat: (Karawang) Diagnosa Dokter: (Panas) Dokter: (Dr. Alpa) Kode obat: (OBT0001) (OBT0002) Nama Obat: (paracetamol) (Amoxilin)	Sistem akan menerima dan menampilkan pesan “Berhasil Menyimpan Data Transaksi”	Sesuai harapan	Valid

		Harga Obat: (12000) (11000) Harga Pemeriksaan: (75000) Total: (98000)			
--	--	--	--	--	--

3.5.3. Spesifikasi *Hardware dan Software*

Tabel III.15.
Spesifikasi *Hardware dan Software*

Kebutuhan	Keterangan
Sistem	Windows 10 64-bit
Processor	Intel 2Core
RAM	2 GB
Harddisk	100 GB
Monitor	14" LCD
Keyboard	102 Keys
Printer	DeskJet
Mouse	Standar
Software	Kebutuhan menjalankan aplikasi secara lokal: Bahasa script programming : Java 8.1 Web server : Apache 3.2 DBMS : MySQL Web Browser : Mozilla firefox