

BAB III

PEMBAHASAN

3.1. Tinjauan Perusahaan

3.1.1. Sejarah Perusahaan

Unit Pelaksanaan Teknis Dinas Daerah (UPTD) Puskesmas Dengan Tempat Perawatan (DTP) Jatisari Kecamatan Jatisari adalah salah satu Puskesmas di kecamatan Jatisari yang merupakan Puskesmas induk dengan luas wilayah 3.490 Km², yang terdiri dari daerah kampung, pesawahan dan sebagian untuk perumahan. Puskesmas Jatisari memiliki puskesmas pembantu (Pustu) yang bertempat di desa Situdam.

Secara kegiatan wilayah kerja UPTD Puskesmas DTP Jatisari Kecamatan Jatisari berbatasan dengan:

1. Sebelah Utara : Puskesmas Cicinde
2. Sebelah Barat : Puskesmas Pacing
3. Sebelah Selatan : Puskesmas Jatisari
4. Sebelah Utara : Puskesmas Patok Beusi Kabupaten Subang

UPTD Puskesmas DTP Jatisari bertempat di Desa Mekarsari Kecamatan Jatisari yang berjarak 30 Km, Kabupaten Karawang, dengan waktu tempuh 1 jam menggunakan roda 4.

UPTD Puskesmas DTP Jatisari mempunyai wilayah kerja terdiri dari 10 Desa, 64 RW dan 182 RT, 16.534 KK, dengan jumlah penduduk laki-laki 29.192 jiwa,

perempuan 26.918 jiwa dengan jarak desa terjauh berjarak 7 Km dari Puskesmas dengan waktu tempuh 20 menit dengan roda 2 dan 30 menit dengan roda 4.

Berdasarkan data proyeksi penduduk tahun 2018 wilayah Puskesmas Jatisari memiliki penduduk :

Berdasarkan tabel 1 diatas dapat dilihat waktu tempuh terlama adalah wilayah desa kalijati yaitu selama 30 menit, keterjangkauan pelayanan kesehatan salah satunya dapat dilihat dari geografis wilayah dimana wilayah kerja puskesmas jatisari secara geografis dapat digambarkan dari jarak dan waktu ke fasilitas kesehatan dalam hal ini ke Puskesmas Jatisari.

3.1.2. Struktur Organisasi dan Fungsi



Sumber: UPTD Puskesmas DTP Jatisari

Gambar III.1
Struktur Organisasi UPTD Pukesmas DTP Jatisari

Adapun fungsi struktur organisasi UPTD Puskesmas Jatisari sebagai berikut :

1. Kepala Puskesmas

- a. Bertanggung jawab atas seluruh kegiatan Puskesmas.
- b. Memimpin pelaksanaan tugas pokok Puskesmas dan fungsi Puskesmas.
- c. Membina kerjasama karyawan atau karyawan dalam pelaksanaan tugas sehari-hari.
- d. Melakukan pengawasan menyeluruh bagi seluruh pelaksanaan kegiatan program dan pengelolaan keuangan.
- e. Memeriksa dan membuat laporan keuangan Puskesmas dengan dibantu oleh bendahara.

2. Bendahara

- a. Menerima dan menginput keuangan.
- b. Menginput keuangan untuk semua pengeluaran maupun penerimaan Puskesmas.
- c. Membuat laporan keuangan penerimaan pengambilan setoran dan pengeluaran Puskesmas.
- d. Membantu Kepala Puskesmas dalam membuat laporan keuangan Puskesmas khususnya pendapatan jasanya.

3. Fungsi Petugas Loker Pendaftaran

- a. Mendaftar pasien yang datang berobat.
- b. Mencatat di register.
- c. Mengisi identitas pasien di kartu rawat jalan atau kertas pendaftaran.
- d. Mengantar Rekam Medis pasien rawat jalan ke ruang BP.
- e. Bertanggung jawab atas pengeluaran kertas antrian.

- f. Menyusun Rekam Medis rawat jalan pasien pada rak status sesuai urutan nomor kode.
- g. Membantu merencanakan kebutuhan Rekam Medis rawat jalan, resep, dan kartu tanda pengenal.
- h. Mencatat register baru atau lama, register bayar atau gratis.

4. Admin

- a. Bertanggung jawab atas penerimaan uang retribusi.
- b. Mencatat hasil penerimaan retribusi di buku bantu.
- c. Menghitung kwitansi pembayaran masuk dan setoran harian.
- d. Menyetorkan kepada bendahara penerimaan hasil penerimaan retribusi setiap hari.
- e. Membuat laporan retribusi pendapatan jasa tindakan medis laboratorium.

5. Petugas Laboratorium

- a. Membuat perencanaan kebutuhan alat atau sarana, reagensia laboratorium dan bahan habis pakai lainnya yang dibutuhkan selama 1 tahun.
- b. Membantu perencanaan pengembangan kegiatan laboratorium.
- c. Melaksanakan kegiatan pemeriksaan laboratorium sesuai prosedur.
- d. Melaksanakan pencatatan kegiatan laboratorium.
- e. Membantu admin dalam pembuatan laporan pendapatan laboratorium.

3.1.3. Visi dan Misi UPTD Puskesmas DTP Jatisari

Pada dasarnya visi dan misi UPTD Puskesmas DTP Jatisari adalah meningkatkan kualitas sumber daya kesehatan pada masyarakat agar kualitas sumber daya manusianya menjadi masyarakat dengan gaya hidup yang sehat, dengan demikian

secara garis besar maka visi dan misi UPTD Puskesmas DTP Jatisari adalah sebagai berikut:

Visi

Terwujudnya masyarakat jatisari sehat dan mandiri.

Misi

1. Meningkatkan kualitas dan jaminan pelayanan kesehatan.
2. Meningkatkan pemberdayaan masyarakat dan kesehatan keluarga.
3. Meningkatkan pencegahan, pengendalian penyakit dan penyehatan lingkungan.
4. Meningkatkan promosi kesehatan yang berdaya guna sehingga mendorong masyarakat untuk bergaya hidup sehat.
5. Meningkatkan kualitas sumber daya kesehatan.

3.2. Tinjauan Kasus

3.2.1. Proses Bisnis Sistem Berjalan

Proses Bisnis Sistem Berjalan pada UPTD Puskesmas DTP Jatisari dalam melakukan proses pencatatan pendapatan jasa tindakan medis laboratorium dan proses pembuatan laporan pendapatan tindakan medis laboratorium. Adapun langkah-langkahnya antara lain:

1. Prosedur Pendaftaran Pasien

Prosedur ini diawali dengan pasien atau keluarga pasien yang datang ke Puskesmas untuk berobat yang melakukan pendaftaran di loket pendaftaran. Kemudian petugas pendaftaran meminta kartu pasien kemudian jika pasien mendaftarkan diri pada poli klinik umum, maka pasien akan didaftarkan oleh

petugas pendaftaran dan akan melakukan pemeriksaan oleh dokter. Setelah melakukan pemeriksaan oleh dokter, selanjutnya pasien diminta mendaftarkan diri untuk melakukan pemeriksaan laboratorium dengan diberikannya surat rujukan pemeriksaan laboratorium oleh dokter untuk dapat diketahui lebih jelas penyakitnya. Kemudian pasien akan menuju loket pendaftaran, disini juga terdapat pasien lain yang dapat mendaftarkan dirinya untuk pemeriksaan laboratorium tanpa harus melakukan pemeriksaan dokter terlebih dahulu. Selanjutnya petugas pendaftaran menanyakan pemeriksaan laboratorium tujuan pasien, setelah itu petugas pendaftaran membuat kwitansi pembayaran pemeriksaan laboratorium dengan daftar harga pemeriksaan laboratorium yang telah ditentukan. Kemudian petugas pendaftaran mencatat data kunjungan pasien kedalam buku catatan registrasi kunjungan pasien. Setelah itu petugas pendaftaran akan memberikan kwitansi pembayaran pemeriksaan laboratorium kepada admin, admin kemudian meminta uang pembayaran pemeriksaan laboratorium kepada pasien. Pasien membayar sejumlah uang sesuai dengan kwitansi pembayaran pemeriksaan laboratorium. Setelah menerima uang, admin menstempel kwitansi pembayaran tersebut dan mencatat pada buku catatan pendapatan jasa laboratoriumnya kemudian admin memberikan salinan kwitansi pembayaran kepada pasien, selanjutnya pasien akan melakukan pemeriksaan laboratorium.

2. Prosedur Pemeriksaan Pasien

Setelah pasien melakukan pendaftaran, pasien menuju ruangan laboratorium dan menyerahkan kwitansi pembayarannya untuk diambil spesimen. Petugas laboratorium kemudian mencatat registrasi pemeriksaan laboratorium pada buku catatan transaksi laboratorium. Lalu pasien menuju ruang pengambilan spesimen

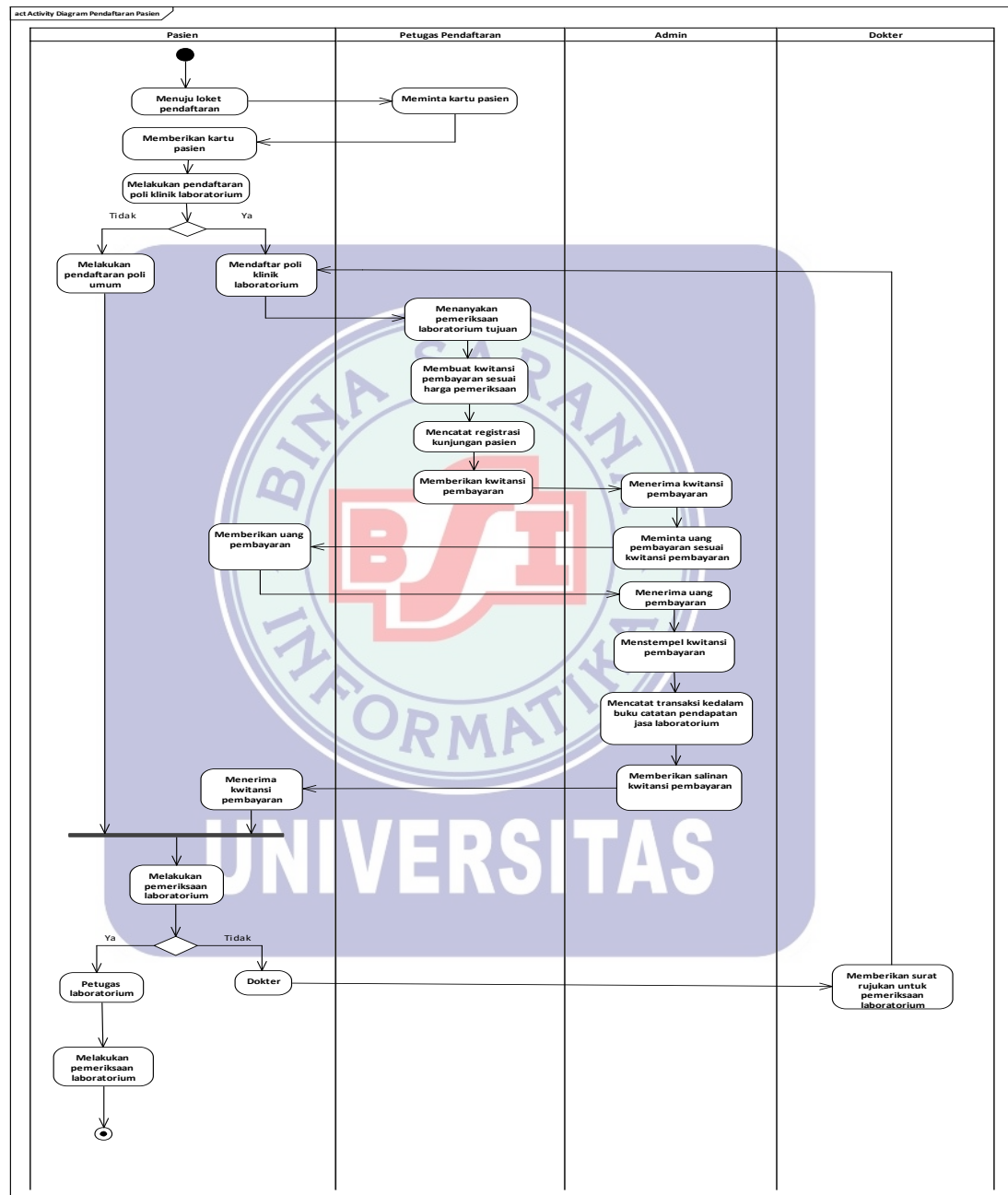
setelah dipanggil oleh petugas laboratorium, petugas laboratorium kemudian mengambil spesimen pasien. Setelah pengambilan spesimen, petugas laboratorium memeriksa hasil spesimen tersebut. Jika spesimen pemeriksaan laboratorium tersebut langsung mengeluarkan hasil maka petugas laboratorium akan menyerahkan formulir hasil pemeriksaan tersebut kepada pasien. Kemudian pasien menyerahkan formulir hasil pemeriksaan tersebut kepada dokter untuk membacakan hasil pemeriksaannya. Jika hasil spesimen pemeriksaan laboratorium tersebut menunggu beberapa hari, maka pasien akan menunggu beberapa hari.

3. Prosedur Pembuatan Laporan Pendapatan Jasa Laboratorium

Admin mengumpulkan semua bukti-bukti transaksi berupa kwitansi pembayaran laboratorium selama satu bulan yang disuaikan dengan buku catatan pendapatan jasa laboratoriumnya. Kemudian admin membuat laporan pendapatan jasa laboratorium dengan dibantu petugas laboratorium dengan cara petugas laboratorium mengecek kesamaan antara data dilaporan yang dibuat admin dengan buku catatan transaksi laboratorium yang dimiliki petugas laboratorium. Selanjutnya hasil laporan pendapatan medis laboratorium diserahkan kepada bendahara. Selanjutnya bendahara mengecek dan meberikannya hasil laporan pendapatan tindakan medis laboratorium kepada kepala puskesmas untuk diperiksa bersama sehingga dapat diperoleh hasil naik atau turunnya pendapatan pada tindakan medis laboratorium. Setelah itu kepala puskesmas akan menandatangani laporan pendapatan jasa laboratoriumnya. Kemudian laporan pendapatan jasa laboratorium akan disimpan oleh bendahara sebagai arsip.

3.2.2. Activity Diagram

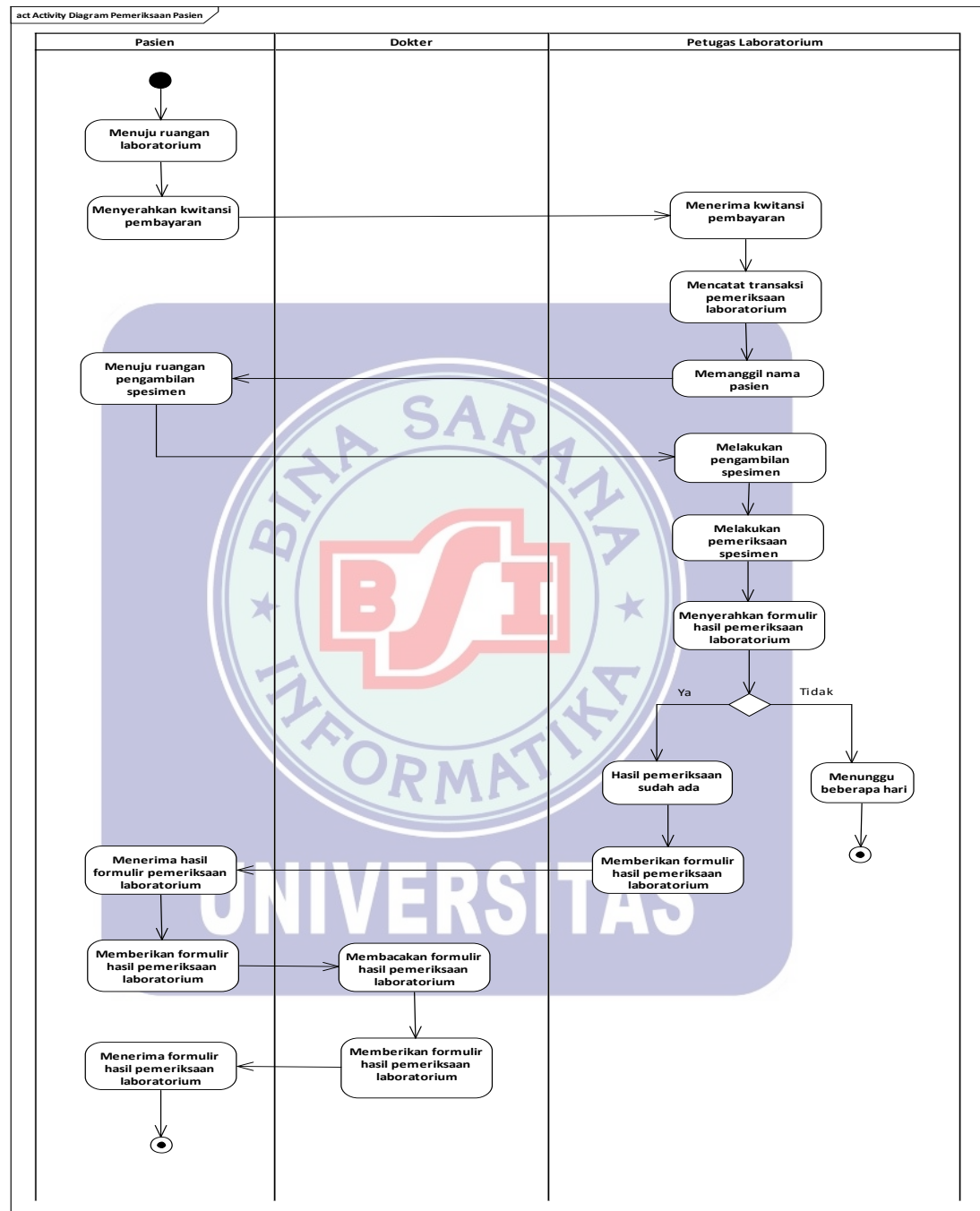
1. Berikut adalah gambaran rangkaian aktivitas sistem bisnis berjalan pendaftaran pasien:



Sumber: Hasil penelitian (2020).

Gambar III.2
Activity Diagram Pendaftaran Pasien

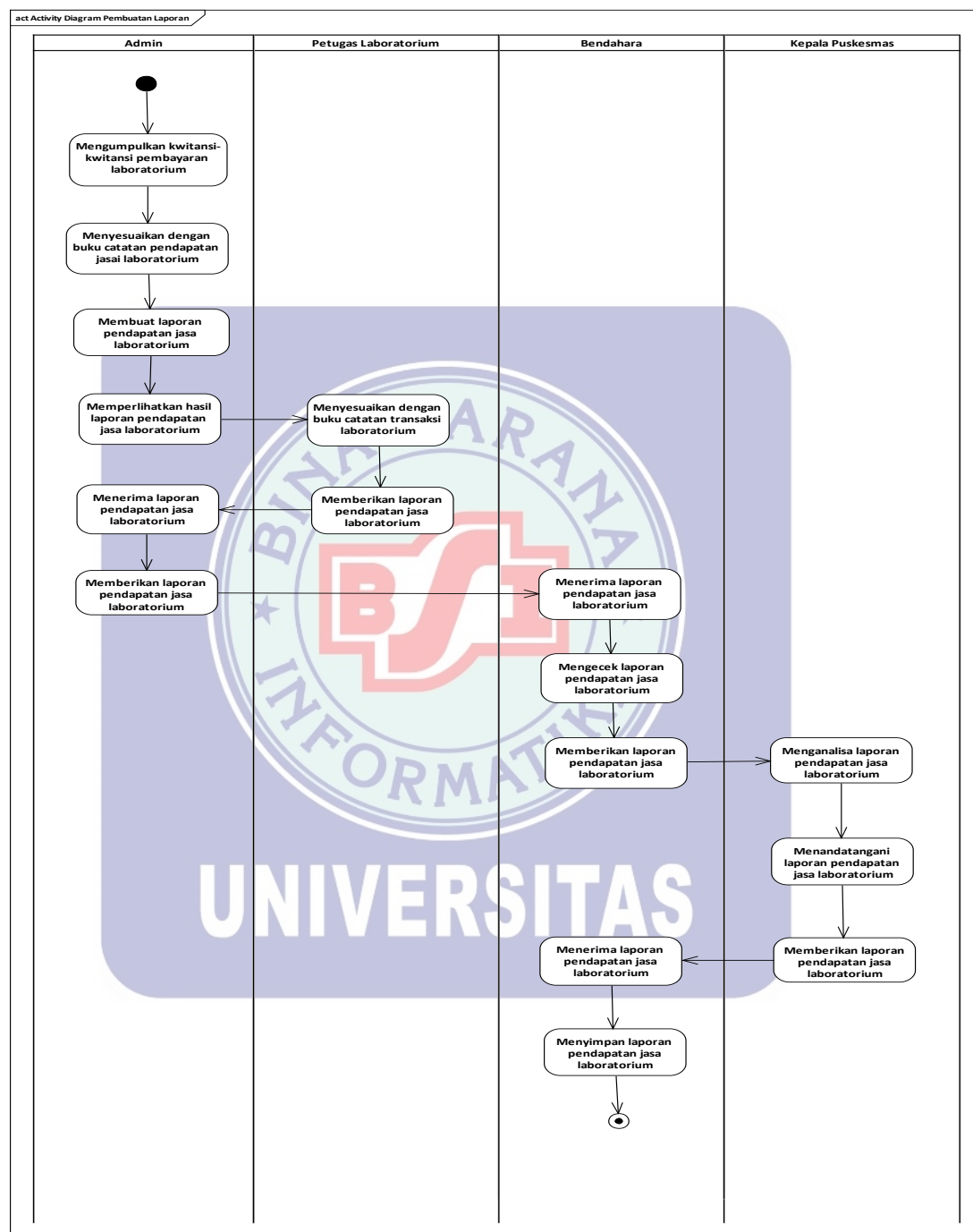
2. Berikut adalah gambaran rangkaian aktivitas sistem bisnis berjalan pemeriksaan pasien:



Sumber: Hasil Penelitian (2020).

Gambar III.3
Activity Diagram Pemeriksaan Pasien

3. Berikut adalah gambaran rangkaian aktivitas sistem bisnis berjalan pembuatan laporan pendapatan jasa laboratorium:



Sumber: Hasil Penelitian (2020).

Gambar III.4
Activity Diagram Pembuatan Laporan

3.2.3. Dokumen Masukan

Dokumen masukan diartikan sebagai dokumen yang digunakan sebagai dasar memperoleh data-data yang nantinya akan diproses untuk menghasilkan suatu keluaran yang disebut dengan *output*.

Rincian dokumen masukan sebagai berikut:

1. Nama Dokumen: Daftar Harga

Fungsi : Sebagai patokan harga pemeriksaan laboratorium
 Sumber : UPTD Puskesmas DTP Jatisari
 Tujuan : Data untuk pembuatan kwitansi pembayaran laboratorium
 Media : Kertas
 Jumlah : 1 lembar
 Frekuensi : Setiap ada pendaftaran tindakan medis laboratorium
 Format : Lampiran A.1.

2. Nama Dokumen: Kartu Pasien

Fungsi : Sebagai Data Pencatatan Kunjungan Pasien
 Sumber : Pasien
 Tujuan : Data untuk pembuatan kwitansi pembayaran laboratorium
 Media : Kertas
 Jumlah : 1 lembar
 Frekuensi : Setiap ada pendaftaran tindakan medis laboratorium
 Format : Lampiran A.2.

3. Nama Dokumen: Buku Catatan Registrasi Kunjungan Pasien

Fungsi : Sebagai hasil pencatatan data kunjungan pasien
 Sumber : Petugas Pendaftaran

Tujuan : Sebagai catatan pembantu pencatatan transaksi laboratorium
 Media : Buku
 Jumlah : 1 Buah
 Frekuensi : Setiap ada pendaftaran pasien
 Format : Lampiran A.3.

4. Nama Dokumen: Buku Catatan Transaksi Laboratorium

Fungsi : Sebagai hasil pencatatan transaksi laboratorium
 Sumber : Admin
 Tujuan : Untuk data pembuatan laporan pendapatan jasa laboratorium
 Media : Buku
 Jumlah : 1 Buah
 Frekuensi : Setiap ada pendaftaran tindakan medis laboratorium
 Format : Lampiran A.4.

5. Nama Dokumen : Buku Catatan Laboratorium

Fungsi : Sebagai hasil pencatatan transaksi laboratorium
 Sumber : Petugas laboratorium
 Tujuan : Data pembuatan laporan pendapatan jasa laboratorium
 Media : Buku
 Jumlah : 1 Buah
 Frekuensi : Setiap ada pemeriksaan laboratorium
 Format : Lampiran A.5.

6. Nama Dokumen : Surat Permintaan Pemeriksaan Laboratorium

Fungsi : Sebagai alat untuk membuat kwitansi pembayaran
 Sumber : Dokter
 Tujuan : Petugas Pendaftaran- Petugas Laboratorium

Media	: Kertas
Jumlah	: 1 Lembar
Frekuensi	: Setiap ada pemeriksaan laboratorium
Format	: Lampiran A.6.

3.2.4. Dokumen Keluaran

Dokumen keluaran diartikan sebagai data-data yang telah di olah yang akan mendukung kegiatan manajemen serta merupakan dokumen dari catatan hasil laporan.

Rincian dokumen keluaran sebagai berikut:

- Nama Dokumen** : Kwitansi Pembayaran

Fungsi : Sebagai bukti pembayaran tindakan medis laboratorium

Sumber : Admin

Tujuan : Data pembuatan laporan pendapatan jasa laboratorium

Media : Kertas

Jumlah : 1 Lembar

Frekuensi : Setiap ada pendaftaran tindakan medis laboratorium

Format : Lampiran B.1.
- Nama Dokumen** : Formulir Hasil Pemeriksaan Laboratorium

Fungsi : Sebagai bukti hasil pemeriksaan tindakan medis laboratorium

Sumber : Petugas Laboratorium

Tujuan : Pasien-Dokter

Media : Kertas

Jumlah : 1 lembar

Frekuensi : Setiap ada pemeriksaan laboratorium

Format : Lampiran B.2.

3. Nama Dokumen : Laporan Pendapatan Jasa Laboratorium

Fungsi : Sebagai hasil laporan pendapatan jasa laboratorium

Sumber : Admin

Tujuan : Bendahara-Kepala Puskesmas

Media : Kertas

Jumlah : 1 lembar

Frekuensi : Setiap bulan

Format : Lampiran B.3.

3.2.5. Permasalahan Pokok

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat di simpulkan permasalahan pada UPTD Puskesmas DTP Jatisari sebagai berikut:

1. UPTD Puskesmas DTP Jatisari sama sekali belum menggunakan sistem informasi akuntansi pada sistem Pendapatan Jasa Medis Laboratorium. Metode pencacatan keuangan dilakukan secara manual atau masih ditulis tangan dan hanya menggunakan buku catatan sehingga kurang efektif dan efisien.
2. Pencarian data yang dilakukan menjadi sulit karena penyimpanan masih belum rapi dan mengakibatkan pencampuran data sehingga memakan waktu yang lama.
3. Keamanan data yang kurang terjamin sehingga data dapat di akses oleh siapapun.
4. Pembuatan laporan pendapatan jasa medis laboratorium dilakukan dengan ditulis tangan.

3.2.6. Pemecahan Masalah

Untuk menutupi permasalahan yang ada pada UPTD Puskesmas DTP Jatisari, penulis memiliki pemecahan masalah dengan cara memakai sistem yang sudah terkomputerisasi. Dengan sistem yang sudah terkomputerisasi ini mempermudah proses pencatatan maupun pembuatan laporan keuangan khususnya pada sistem Pendapatan Jasa Tindakan Medis. Dan data-data dapat disimpan dalam database, sehingga apabila dibutuhkan laporan dapat dicetak kapanpun. Dari permasalahan yang ada maka penulis membuat aplikasi untuk sistem informasi pendapatan jasa tindakan medis laboratorium berbasis desktop menggunakan bahasa pemrograman java dengan Netbeans.

3.3. Analisa Kebutuhan Software

Kebutuhan Software pada Pengolahan Data Pendapatan Jasa Medis Laboratorium pada UPTD Puskesmas DTP Jatisari akan menjadi kebutuhan pengguna dalam penggunaan aplikasi berbasis Desktop yang telah dirancang.

3.3.1. Analisa Kebutuhan

1. Admin

- A.1. Admin dapat *login*.
- A.2. Admin dapat mengolah data pasien.
- A.3. Admin dapat mengelola data pemeriksaan laboratorium.
- A.4. Admin dapat mengelola data akun.
- A.5. Admin dapat mengolah data transaksi laboratorium.
- A.6. Admin dapat mengelola data jurnal pendapatan jasa laboratorium.

A.7. Admin dapat membuat laporan, yaitu laporan pendapatan dan laporan jurnal

A.8. Admin dapat *logout*.

2. Kepala Puskesmas

B.1. Kepala Puskesmas dapat *login*.

B.2. Kepala Puskesmas dapat mengelola data user

B.3. Kepala Puskesmas dapat melihat dan mencetak laporan, yaitu laporan pendapatan dan laporan jurnal pendapatan jasa laboratorium.

B.4. Kepala Puskesmas dapat *logout*.

3. Bendahara

C.1. Bendahara dapat *login*.

C.2. Bendahara dapat melihat dan mencetak laporan, yaitu laporan pendapatan dan laporan jurnal pendapatan jasa laboratorium.

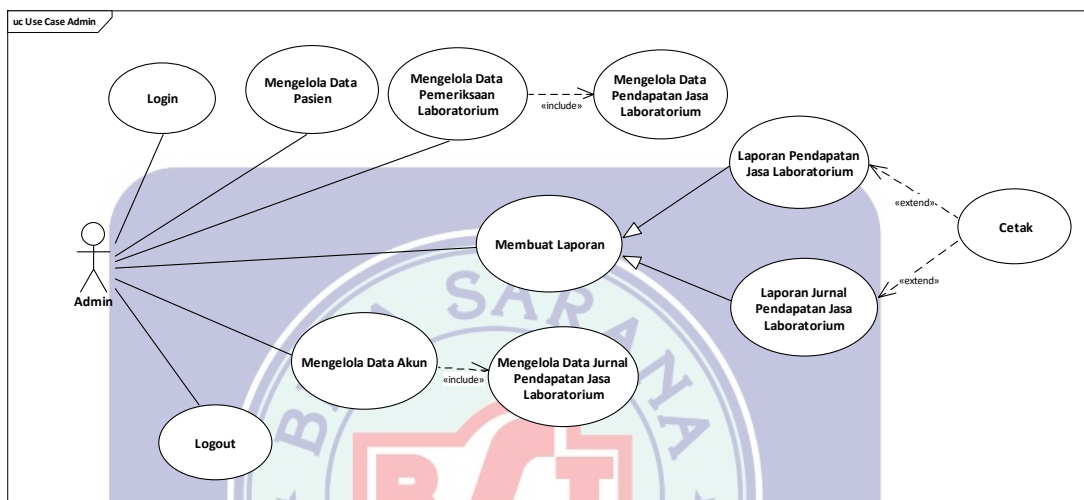
C.3. Bendahara dapat *logout*.



3.3.2. Use Case Diagram

1. Use Case Diagram Admin

Berikut merupakan *Use Case Diagram* bagian admin yang terdapat pada analisa kebutuhan yaitu:



Sumber: Hasil Rancangan (2020).

Gambar III.5
Use Case Diagram Admin

Tabel III.2

Deskripsi Use Case Diagram Halaman Login Admin

Use Case Halaman Login	
Tujuan	Mengijinkan admin untuk masuk kehalaman login dan memasukkan id user dan password.
Aktor	Admin
Kondisi Awal	Menampilkan halaman login.
Skenario Utama	- Admin membuka sistem. - Sistem menampilkan halaman login, untuk memasukkan id user dan password.
Skenario Alternatif	Jika salah satu data tidak di isi atau salah, maka sistem akan menunjukkan pesan “Id User Keliru Atau Password Yang Anda Masukkan Keliru”.
Kondisi Akhir	Sistem menampilkan menu utama bendahara.

Tabel III.3

Deskripsi Use Case Diagram Mengelola Data Pasien

<i>Use Case Mengelola Data Pasien</i>	
Tujuan	Mengijinkan admin untuk mengelola data <i>user</i> .
Aktor	Admin
Kondisi Awal	Login tervalidasi dan valid
Skenario Utama	1. Admin memilih form data pasien. 2. Sistem menampilkan data pasien, untuk diolah data pasiennya.
Skenario Alternatif	1. Jika data pasien belum lengkap, maka sistem akan menunjukkan pesan “ Mohon Isi Data Dahulu”. 2. Jika format nama yang diisi salah, maka sistem akan menunjukkan pesan “Format [Nama Pasien] Tidak Valid”. 3. Jika data yang dipilih gagal diubah, maka sistem akan menunjukkan pesan “Data Gagal Diubah”. 4. Jika data yang dipilih gagal dihapus, maka sistem akan menunjukkan pesan “Data Gagal Dihapus”.
Kondisi Akhir	Sistem menampilkan data pasien.

Tabel III.4

Deskripsi Use Case Diagram Mengelola Data Pemeriksaan Laboratorium

<i>Use Case Mengelola Data Pemeriksaan Laboratorium</i>	
Tujuan	Mengijinkan admin untuk mengelola data pemeriksaan laboratorium.
Aktor	Admin
Kondisi Awal	Login tervalidasi dan valid
Skenario Utama	1. Admin memilih form data pemeriksaan laboratorium. 2. Sistem menampilkan data pemeriksaan laboratorium, untuk diolah datanya.
Skenario Alternatif	1. Jika data pemeriksaan laboratorium belum lengkap, maka sistem akan menunjukkan pesan “ Mohon Isi Data Dahulu”. 2. Jika format nama yang diisi salah, maka sistem akan menunjukkan pesan “Format [Nama Pemeriksaan] Tidak Valid”. 3. Jika data yang dipilih gagal diubah, maka sistem akan menunjukkan pesan “Data Gagal Diubah”. 4. Jika data yang dipilih gagal dihapus, maka sistem akan menunjukkan pesan “Data Gagal Dihapus”.
Kondisi Akhir	Sistem menampilkan data pemeriksaan laboratorium.

Tabel III.5

Deskripsi Use Case Diagram Mengelola Data Akun

<i>Use Case Mengelola Data Akun</i>	
Tujuan	Mengijinkan admin untuk mengelola data akun.
Aktor	Admin
Kondisi Awal	Login tervalidasi dan valid
Skenario Utama	1. Admin memilih form data akun. 2. Sistem menampilkan data akun, untuk diolah datanya.
Skenario Alternatif	1. Jika data akun belum lengkap, maka sistem akan menunjukkan pesan “ Mohon Isi Data Dahulu”. 2. Jika format nama yang diisi salah, maka sistem akan menunjukkan pesan “Format [Nama Akun] Tidak Valid”. 3. Jika data yang dipilih gagal diubah, maka sistem akan menunjukkan pesan “Data Gagal Diubah”. 4. Jika data yang dipilih gagal dihapus, maka sistem akan menunjukkan pesan “Data Gagal Dihapus”.
Kondisi Akhir	Sistem menampilkan data akun.

Tabel III.6

Deskripsi Use Case Diagram Mengelola Data Pendapatan

<i>Use Case Mengelola Data Pendapatan</i>	
Tujuan	Mengijinkan admin untuk mengelola pendapatan laboratorium.
Aktor	Admin
Kondisi Awal	Login tervalidasi dan valid
Skenario Utama	1. Admin memilih form data pendapatan. 2. Sistem menampilkan data pendapatan laboratorium, untuk diolah datanya.
Skenario Alternatif	1. Jika data pasien belum dipilih, maka sistem akan menunjukkan pesan “ Pasien Belum Dipilih”. 2. Jika data pemeriksaan laboratorium belum dipilih, maka sistem akan menunjukkan pesan “Belum Ada Pemeriksaan Laboratorium Yang Dipilih”.
Kondisi Akhir	Sistem menampilkan data pendapatan laboratorium.

Tabel III.7

Deskripsi Use Case Diagram Mengelola Data Jurnal Pendapatan Laboratorium

<i>Use Case Mengelola Data Jurnal Pendapatan Laboratorium</i>	
Tujuan	Mengijinkan admin untuk mengelola jurnal pendapatan laboratorium.
Aktor	Admin
Kondisi Awal	Login tervalidasi dan valid
Skenario Utama	1. Admin memilih form data jurnal pendapatan laboratorium. 2. Sistem menampilkan data jurnal pendapatan laboratorium, untuk diolah datanya.
Skenario Alternatif	1. Jika keterangan belum diisi, maka sistem akan menunjukkan pesan “Keterangan Belum Diisi”. 2. Jika data akun belum dipilih, maka sistem akan menunjukkan pesan “Belum Ada Akun Yang Dipilih”.
Kondisi Akhir	Sistem menampilkan data jurnal pendapatan laboratorium.

Tabel III.8

Deskripsi Use Case Diagram Membuat Laporan Jurnal Pendapatan Jasa Laboratorium

<i>Use Case Membuat Laporan Jurnal Pendapatan Jasa Laboratorium</i>	
Tujuan	Mengijinkan admin untuk melihat atau mencetak laporan jurnal pendapatan jasa laboratorium.
Aktor	Admin
Kondisi Awal	Login tervalidasi dan valid
Skenario Utama	1. Admin membuka form laporan jurnal pendapatan jasa laboratorium, dan memasukkan periode tanggal yang akan dicetak. 2. Sistem menampilkan data laporan jurnal pendapatan jasa laboratorium, untuk dilihat maupun dicetak.
Skenario Alternatif	1. Jika mengklik cetak, maka sistem akan menampilkan data laporan jurnal pendapatan jasa laboratorium. 2. Jika keluar, maka sistem akan keluar.
Kondisi Akhir	Sistem menampilkan form laporan jurnal pendapatan jasa laboratorium.

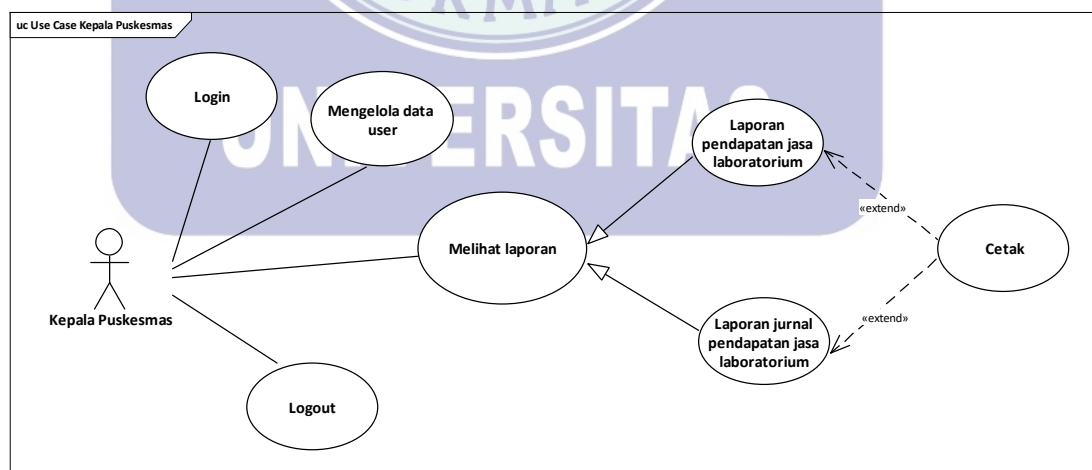
Tabel III.9

Deskripsi Use Case Diagram Membuat Laporan Jurnal Pendapatan Jasa Laboratorium

Use Case Membuat Laporan Pendapatan Jasa Laboratorium	
Tujuan	Mengijinkan admin untuk melihat atau mencetak laporan pendapatan jasa laboratorium.
Aktor	Admin
Kondisi Awal	Login tervalidasi dan valid
Skenario Utama	1. Admin membuka form laporan pendapatan jasa laboratorium, dan memasukkan periode tanggal yang akan dicetak. 2. Sistem menampilkan data laporan pendapatan jasa laboratorium, untuk dilihat maupun dicetak.
Skenario Alternatif	3. Jika mengklik cetak, maka sistem akan mencetak data laporan pendapatan jasa laboratorium. 4. Jika keluar, maka sistem akan keluar.
Kondisi Akhir	Sistem menampilkan form laporan pendapatan jasa laboratorium.

2. *Use Case Diagram Kepala Puskesmas*

Berikut merupakan *Use Case Diagram* bagian kepala puskesmas yang terdapat pada analisa kebutuhan yaitu:



Sumber: Hasil Rancangan (2020).

Gambar III.6

Use Case Diagram Kepala Puskesmas

Tabel III.10**Deskripsi Use Case Diagram Halaman Login Kepala Puskesmas**

<i>Use Case</i> Halaman Login	
Tujuan	Mengijinkan kepala puskesmas untuk masuk kehalaman login dan memasukkan id user dan password.
Aktor	Kepala Puskesmas
Kondisi Awal	Menampilkan halaman login.
Skenario Utama	1. Kepala puskesmas membuka sistem. 2. Sistem menampilkan halaman login, untuk memasukkan id user dan password.
Skenario Alternatif	1. Jika salah satu data tidak di isi, maka sistem akan menunjukkan pesan “Id User Keliru Atau Password Yang Anda Masukkan Keliru”.
Kondisi Akhir	Sistem menampilkan menu utama kepala puskesmas.

Tabel III.11**Deskripsi Use Case Diagram Mengelola Data User**

<i>Use Case</i> Mengelola Data User	
Tujuan	Mengijinkan kepala puskesmas untuk mengelola data user.
Aktor	Kepala Puskesmas
Kondisi Awal	Login tervalidasi dan valid
Skenario Utama	1. Kepala puskesmas memilih form data user. 2. Sistem menampilkan data user, untuk diolah data usernya.
Skenario Alternatif	1. Jika data user belum lengkap, maka sistem akan menunjukkan pesan “ Mohon Isi Data Dahulu”. 2. Jika format nama yang diisi salah, maka sistem akan menunjukkan pesan “Format [Nama User] Tidak Valid”. 3. Jika data yang dipilih gagal diubah, maka sistem akan menunjukkan pesan “Data Gagal Diubah”. 4. Jika data yang dipilih gagal dihapus, maka sistem akan menunjukkan pesan “Data Gagal Dihapus”.
Kondisi Akhir	Sistem menampilkan data user.

Tabel III.12

Deskripsi Use Case Diagram Mengakses Laporan Jurnal Pendapatan Jasa Laboratorium

<i>Use Case Mengakses Laporan Jurnal Pendapatan Jasa Laboratorium</i>	
Tujuan	Mengijinkan kepala puskesmas untuk melihat atau mencetak laporan jurnal pendapatan jasa laboratorium.
Aktor	Kepala Puskesmas
Kondisi Awal	Login tervalidasi dan valid
Skenario Utama	1. Kepala puskesmas membuka form laporan jurnal pendapatan jasa laboratorium, dan memasukkan periode tanggal yang akan dicetak. 2. Sistem menampilkan data laporan jurnal pendapatan jasa laboratorium, untuk dilihat maupun dicetak.
Skenario Alternatif	1. Jika mengklik cetak, maka sistem akan mencetak laporan jurnal pendapatan jasa laboratorium. 2. Jika keluar, maka sistem akan keluar.
Kondisi Akhir	Sistem menampilkan form laporan jurnal pendapatan jasa laboratorium.

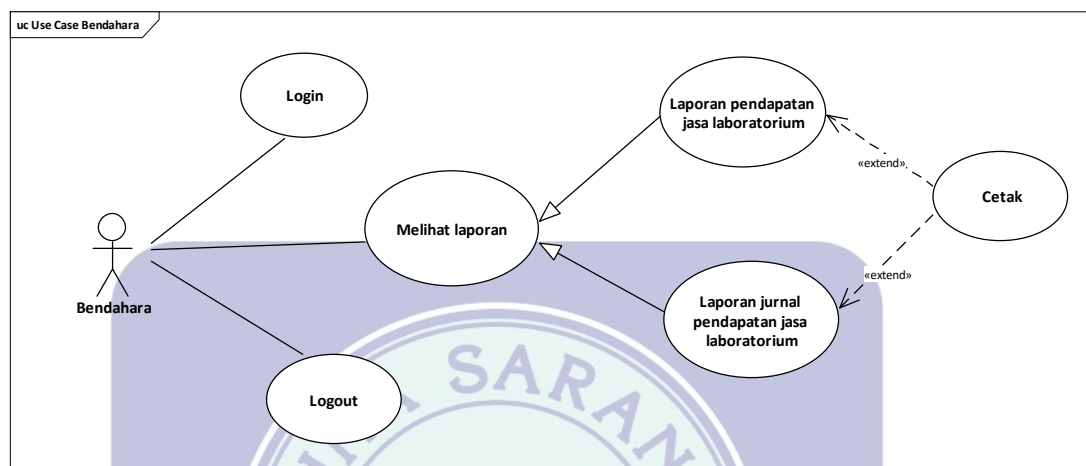
Tabel III.13

Deskripsi Use Case Diagram Mengakses Laporan Pendapatan Jasa Laboratorium

<i>Use Case Mengakses Laporan Pendapatan Jasa Laboratorium</i>	
Tujuan	Mengijinkan kepala puskesmas untuk melihat atau mencetak laporan pendapatan jasa laboratorium.
Aktor	Kepala Puskesmas
Kondisi Awal	Login tervalidasi dan valid
Skenario Utama	1. Kepala puskesmas membuka form laporan pendapatan jasa laboratorium, dan memasukkan periode tanggal yang akan dicetak. 2. Sistem menampilkan data laporan pendapatan jasa laboratorium, untuk dilihat maupun dicetak.
Skenario Alternatif	1. Jika mengklik cetak, maka sistem akan mencetak laporan pendapatan jasa laboratorium. 2. Jika keluar, maka sistem akan keluar.
Kondisi Akhir	Sistem menampilkan form laporan jurnal pendapatan jasa laboratorium.

2. Use Case Diagram Bendahara

Berikut merupakan *Use Case Diagram* bagian bendahara yang terdapat pada analisa kebutuhan yaitu:



Sumber: Hasil Rancangan (2020).

Gambar III.7

Use Case Diagram Bendahara

Tabel III.14

Deskripsi Use Case Diagram Halaman Login Bendahara

Use Case Halaman Login	
Tujuan	Mengijinkan bendahara untuk masuk kehalaman login dan memasukkan id user dan password.
Aktor	Bendahara
Kondisi Awal	Sistem menampilkan halaman login.
Skenario Utama	1. Bendahara membuka sistem. 2. Sistem menampilkan halaman login, untuk memasukkan id user dan password.
Skenario Alternatif	1. Jika salah satu data tidak diisi, maka sistem akan menunjukkan pesan “Id User Keliru Atau Password Yang Anda Masukkan Keliru”.
Kondisi Akhir	Sistem menampilkan menu utama bendahara.

Tabel III.15

Deskripsi Use Case Diagram Mengakses Laporan Jurnal Pendapatan Jasa Laboratorium

<i>Use Case Mengakses Laporan Jurnal Pendapatan Jasa Laboratorium</i>	
Tujuan	Mengijinkan bendahara untuk melihat atau mencetak laporan jurnal pendapatan jasa laboratorium.
Aktor	Bendahara
Kondisi Awal	Login tervalidasi dan valid
Skenario Utama	1. Bendahara membuka form laporan jurnal pendapatan jasa laboratorium, dan memasukkan periode tanggal yang akan dicetak. 2. Sistem menampilkan data laporan jurnal pendapatan jasa laboratorium, untuk dilihat maupun dicetak.
Skenario Alternatif	1. Jika mengklik cetak, maka sistem akan mencetak data laporan jurnal pendapatan jasa laboratorium. 2. Jika keluar, maka sistem akan keluar.
Kondisi Akhir	Sistem menampilkan form laporan jurnal pendapatan jasa laboratorium.

Tabel III.16

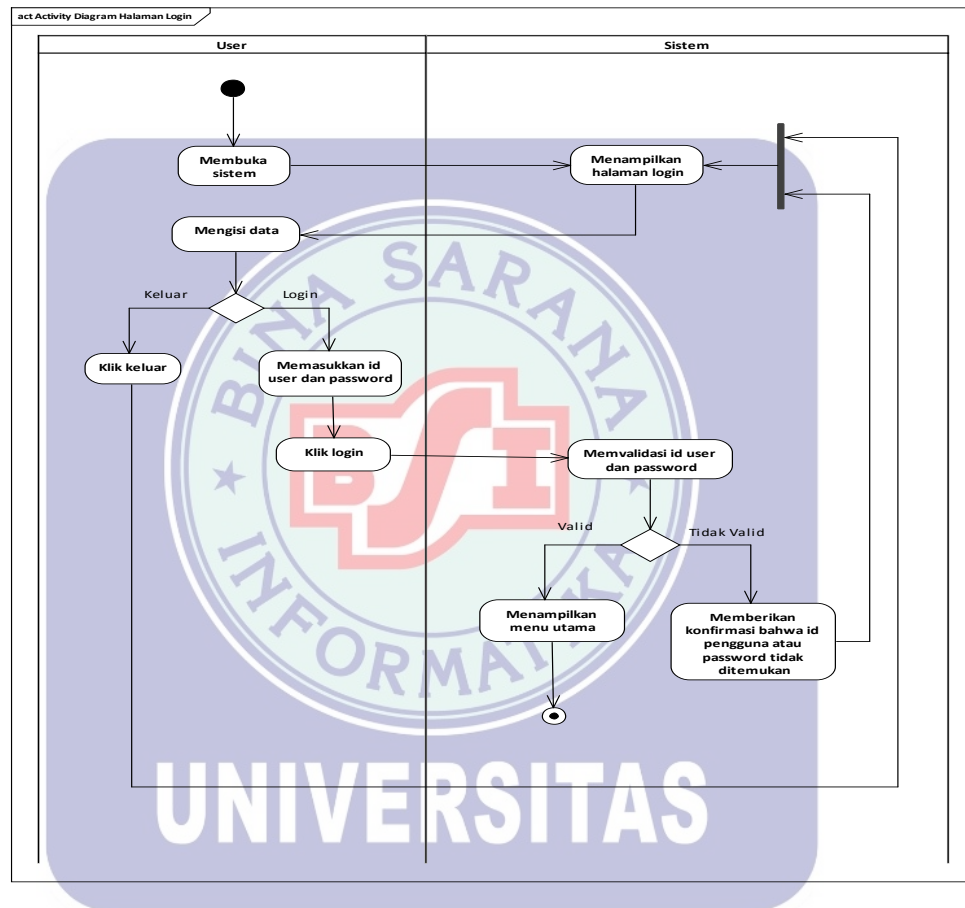
Deskripsi Use Case Diagram Mengakses Laporan Pendapatan Jasa Laboratorium

<i>Use Case Mengakses Laporan Pendapatan Jasa Laboratorium</i>	
Tujuan	Mengijinkan bendahara untuk melihat atau mencetak laporan pendapatan jasa laboratorium.
Aktor	Bendahara
Kondisi Awal	Login tervalidasi dan valid
Skenario Utama	1. Bendahara membuka form laporan pendapatan jasa laboratorium, dan memasukkan periode tanggal yang akan dicetak. 2. Sistem menampilkan data laporan pendapatan jasa laboratorium, untuk dilihat maupun dicetak.
Skenario Alternatif	1. Jika mengklik cetak, maka sistem akan mencetak data laporan pendapatan jasa laboratorium. 2. Jika keluar, maka sistem akan keluar.
Kondisi Akhir	Sistem menampilkan form laporan jurnal pendapatan jasa laboratorium.

3.3.3. Activity Diagram

Berikut ini merupakan *Activity Diagram* dari Sistem Pendapatan Jasa Laboratorium pada UPTD Puskesmas DTP Jatisari.

1. *Activity Diagram* Halaman Login

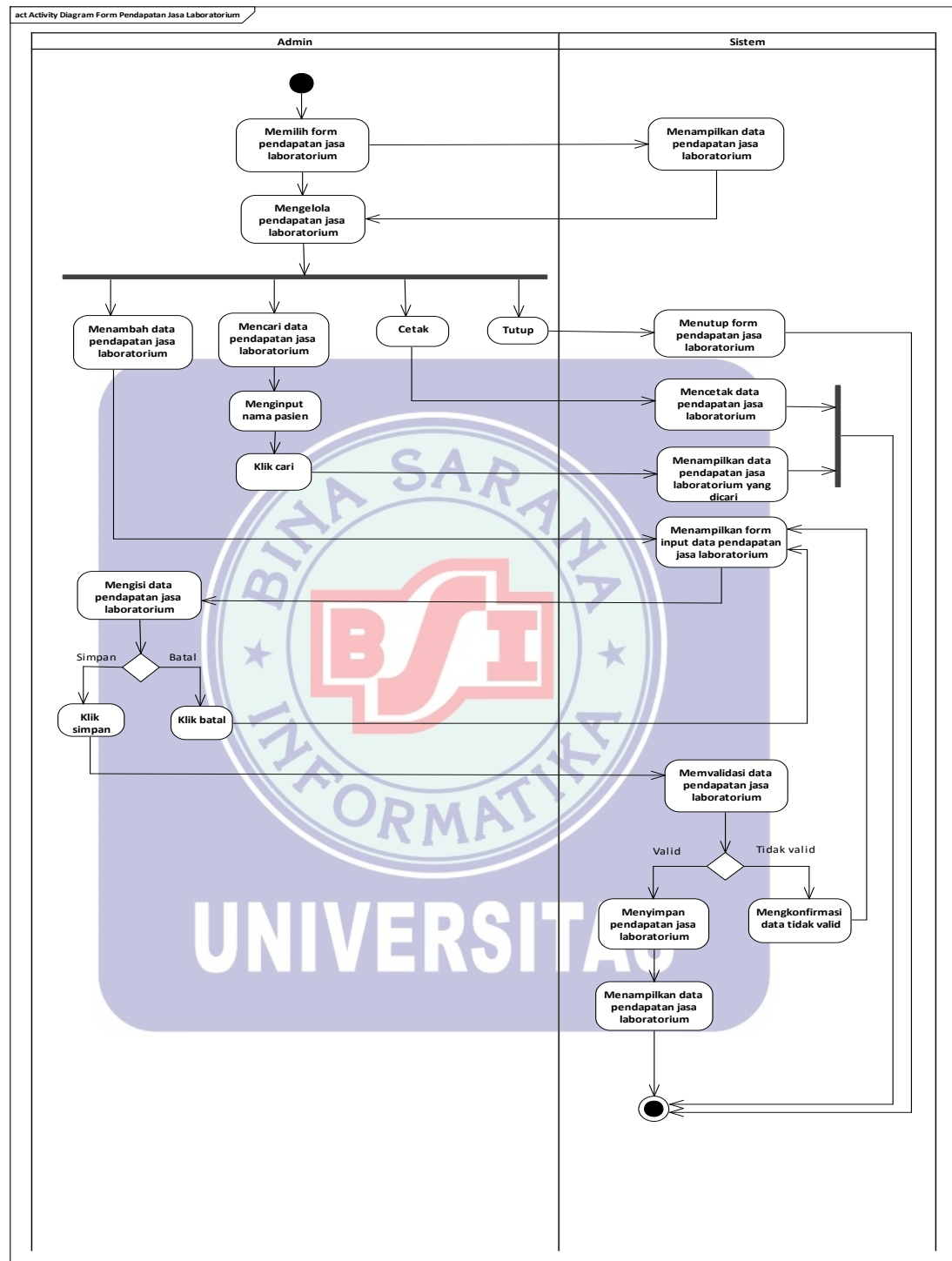


Sumber: Hasil Rancangan (2020).

Gambar III.8

***Activity Diagram* Halaman Login**

3. Activity Diagram Form Data Pendapatan Jasa Laboratorium

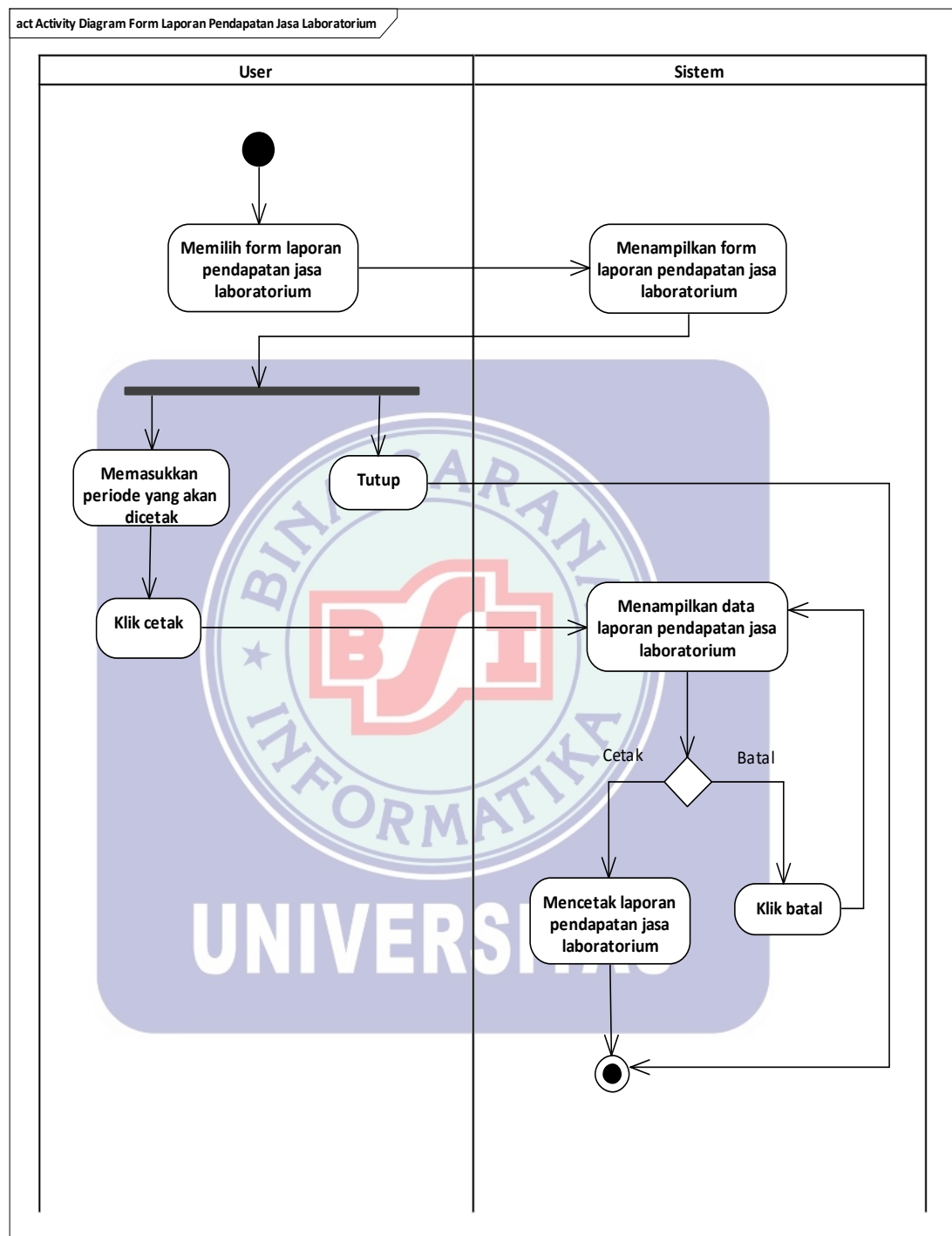


Sumber: Hasil Rancangan (2020).

Gambar III.10

Activity Diagram Form Data Pendapatan Jasa Laboratorium

4. Activity Diagram Halaman Laporan Pendapatan Jasa Laboratorium



Sumber: Hasil Rancangan (2020).

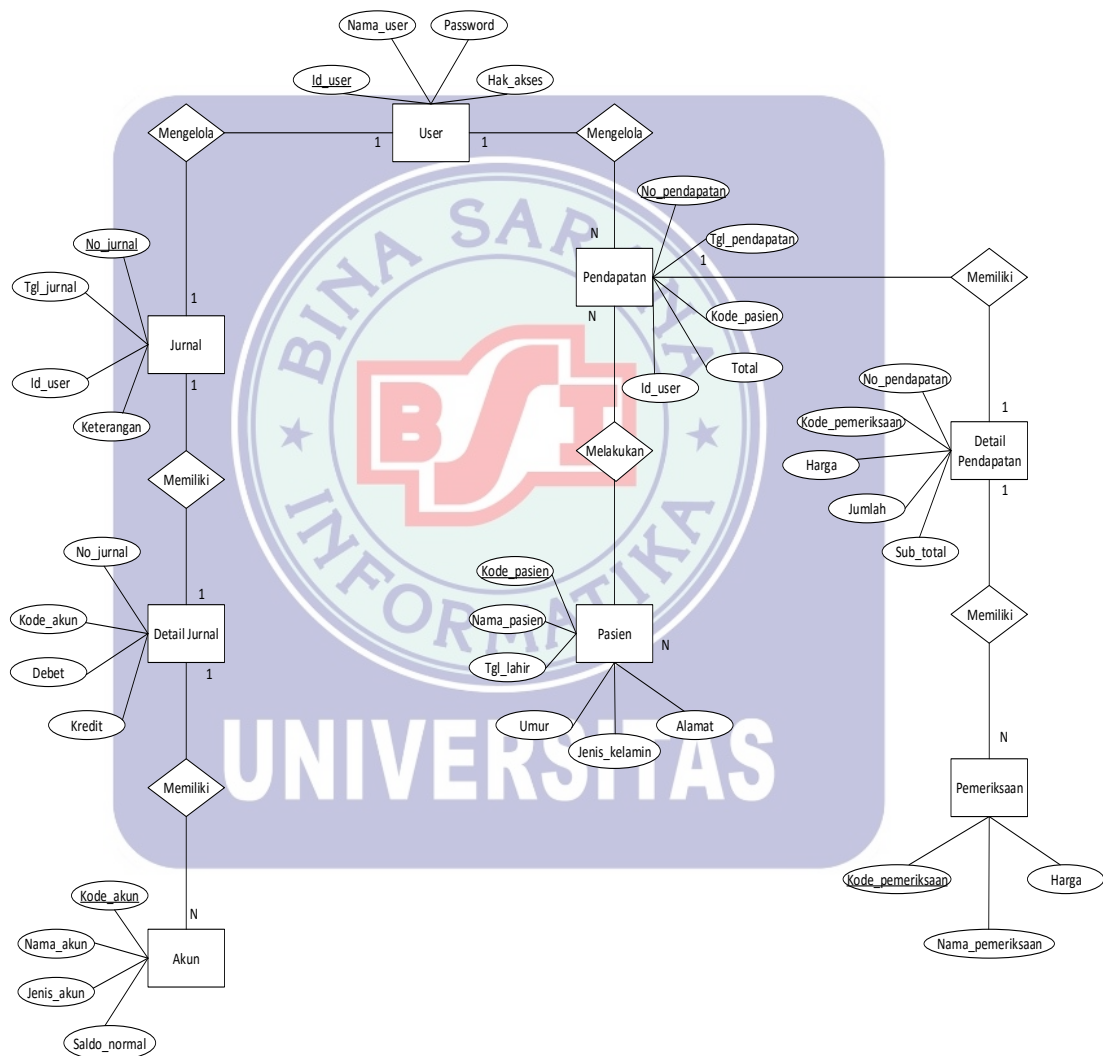
Gambar III.11

Activity Diagram Halaman Laporan Pendapatan Jasa Laboratorium

3.4. Desain

3.4.1. Entity Relationship Diagram (ERD)

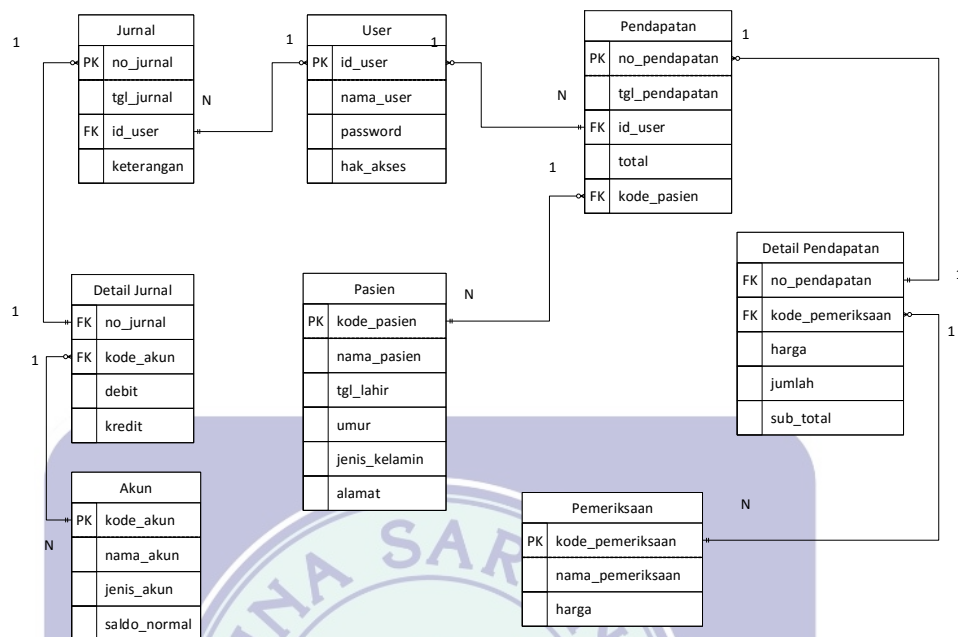
Berikut *Entity Relationship Diagram* (ERD) aplikasi pendapatan jasa laboratorium terdiri dari 8 entitas, dimana ada entitas user, pasien, jasa, akun, transaksi, detail transaksi, jurnal dan detail jurnal.



Gambar III.12

Entity Relationship Diagram (ERD)

3.4.2. Logical Record Structure (LRS)



Sumber: Hasil Rancangan (2020).

Gambar III.13

Logical Record Structure (LRS)

3.4.3. Spesifikasi File

a. Spesifikasi File User

Nama File : User

Akronim : user

Fungsi : Mengelola Data User

Tipe File : Master Data

Organisasi file : Index Sequential

Akses File : Random

Panjang Record : 95

Primary Key : id_user

Software : MySQL

Tabel III.17

Spesifikasi File User

No	Elemen Data	Nama Field	Tipe	Panjang Record	Keterangan
1	Id User	id_user	Char	15	Primary key
2	Nama User	nama_user	Varchar	30	-
3	Password	Password	Varchar	30	-
4	Hak Akses	hak-akses	Varchar	30	-

b. Spesifikasi File Pasien

Nama File : Pasien
 Akronim : pasien
 Fungsi : Mengelola Data Pasien
 Tipe File : Master Data
 Organisasi File : Index Sequential
 Akses File : Random
 Panjang Record : 96
 Primary Key : kode_pasien
 Software : MySQL

Tabel III.18

Spesifikasi File Pasien

No	Elemen Data	Nama Field	Tipe	Panjang Record	Keterangan
1	Kode Pasien	kode_pasien	Char	7	Primary key
2	Nama Pasien	nama_pasien	Varchar	30	-
3	Tanggal Lahir	tgl_lahir	Date	-	-
4	Umur	Umur	Integer	3	-

5	Jenis Kelamin	jenis_kelamin	Varchar	6	-
6	Alamat	Alamat	Varchar	50	-

c. Spesifikasi *File* Pemeriksaan

Nama *File* : Pemeriksaan

Akronim : pemeriksaan

Fungsi : Mengelola Data Pemeriksaan Laboratorium

Tipe *Fille* : *Master Data*

Organisasi *File* : *Index Sequential*

Akses *File* : *Random*

Panjang *Record* : 45

Primary Key : kode_pemeriksaan

Software : MySQL

Tabel III.19

Spesifikasi *File* Pemeriksaan

No	Elemen Data	Nama <i>Field</i>	Tipe	Panjang <i>Record</i>	Keterangan
1	Kode Pemeriksaan	kode_pemeriksaan	Char	5	<i>Primary key</i>
2	Nama Pemeriksaan	nama_pemeriksaan	Varchar	30	-
3	Harga	Harga	Integer	11	-

d. Spesifikasi *File* Akun

Nama *File* : Akun

Akronim : akun

Fungsi : Mengelola Data Akun

Tipe File : *Master Data*
Organisasi File : *Index Sequential*
Akses File : *Random*
Panjang Record : 95
Primary Key : kode_akun
Software : MySQL

Tabel III.20

Spesifikasi *File* Akun

No	Elemen Data	Nama <i>Field</i>	Tipe	Panjang <i>Record</i>	Keterangan
1	Kode Akun	kode_akun	Char	5	Primary key
2	Nama Akun	nama_akun	Varchar	30	-
3	Jenis Akun	jenis_akun	Varchar	30	-
4	Saldo Normal	saldo_normal	Varchar	30	-

e. Spesifikasi *File* Pendapatan

Nama File : Pendapatan
Akronim : pendapatan
Fungsi : Mengelola Pendapatan Laboratorium
Tipe File : Transaksi
Organisasi File : Index Sequential
Akses File : Random
Panjang Record : 23
Primary Key : no_pendapatan
Software : MySQL

Tabel III.21

Spesifikasi *File* Pendapatan

No	Elemen Data	Nama <i>Field</i>	Tipe	Panjang <i>Record</i>	Keterangan
1	Nomor Pendapatan	no_pendapatan	<i>Char</i>	7	<i>Primary key</i>
2	Tanggal Pendapatan	tgl_pendapatan	<i>Date</i>	-	-
3	Kode Pasien	kode_pasien	<i>Char</i>	7	<i>Foreign Key</i>
4	Id User	id_user	<i>Char</i>	15	<i>Foreign Key</i>
5	Total	Total	<i>Integer</i>	11	-

f. Spesifikasi *File* Detail PendapatanNama *File* : Detail Pendapatan

Akronim : detail_pendapatan

Fungsi : Mengelola Data Detail Transaksi

Tipe *File* : TransaksiOrganisasi *File* : *Index Sequential*Akses *File* : *Random*Panjang *Record* : 45*Primary Key* : -*Software* : MySQL

Tabel III.22

Spesifikasi *File* Detail Pendapatan

No	Elemen Data	Nama <i>Field</i>	Tipe	Panjang <i>Record</i>	Keterangan
1	Nomor Pendapatan	no_pendapatan	<i>Char</i>	7	<i>Foreign Key</i>

2	Kode Pemeriksaan	kode_pemeriksaan	Char	5	Foreign Key
4	Harga	Harga	Integer	11	-
5	Jumlah	Jumlah	Integer	11	-
6	Sub Total	sub_total	Integer	11	-

g. Spesifikasi *File Jurnal*

Nama *File* : Jurnal

Akronim : jurnal

Fungsi : Mengelola Jurnal

Tipe *File* : Transaksi

Organisasi *File* : Index Sequential

Akses *File* : Random

Panjang *Record* : 62

Kunci *File* : no_jurnal

Software : MySQL

Tabel III.23

Spesifikasi *File Jurnal*

No	Elemen Data	Nama <i>Field</i>	Tipe	Panjang <i>Record</i>	Keterangan
1	Nomor Jurnal	no_jurnal	Char	7	Primary key
2	Tanggal Jurnal	tgl_jurnal	Date	-	-
4	Id User	id_user	Char	15	Foreign Key
5	Keterangan	Keterangan	Varchar	50	-

h. Spesifikasi *File* Detail Jurnal

Nama *File* : Detail Jurnal

Akronim : detail_jurnal

Fungsi : Mengelola Data Detail Jurnal

Tipe *File* : Transaksi

Organisasi *File* : *Index Sequential*

Akses *File* : *Random*

Panjang *Record* : 34

Primary Key : -

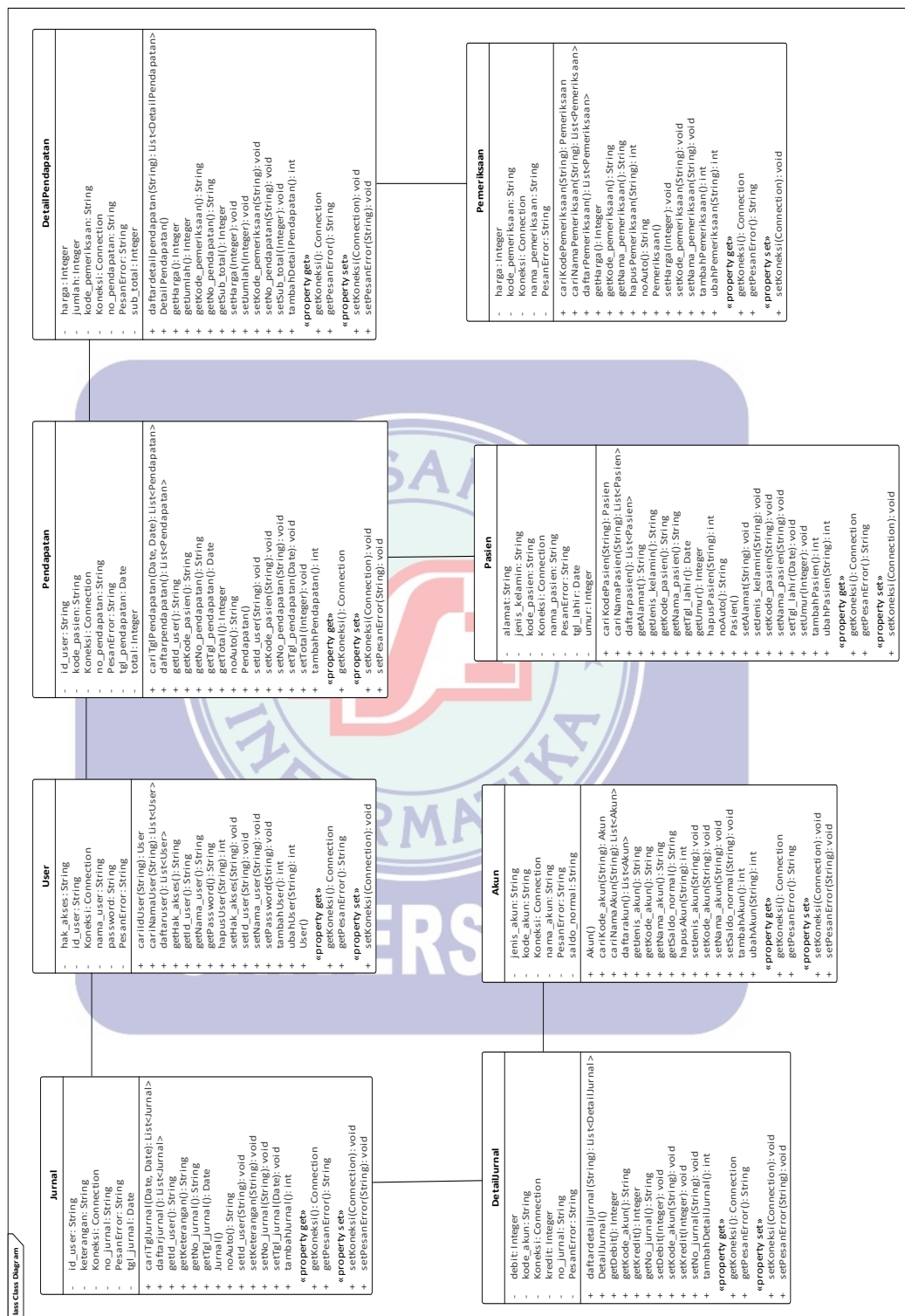
Software : MySQL

Tabel III.24

Spesifikasi *File* Detail Jurnal

No	Elemen Data	Nama <i>Field</i>	Tipe	Panjang <i>Record</i>	Keterangan
1	Nomor Jurnal	no_jurnal	<i>Char</i>	7	<i>Foreign Key</i>
2	Kode Akun	Kode_akun	<i>Char</i>	5	<i>Foreign Key</i>
4	Debit	Debit	<i>Integer</i>	11	-
5	Kredit	Kredit	<i>Integer</i>	11	-

3.4.4. Class Diagram

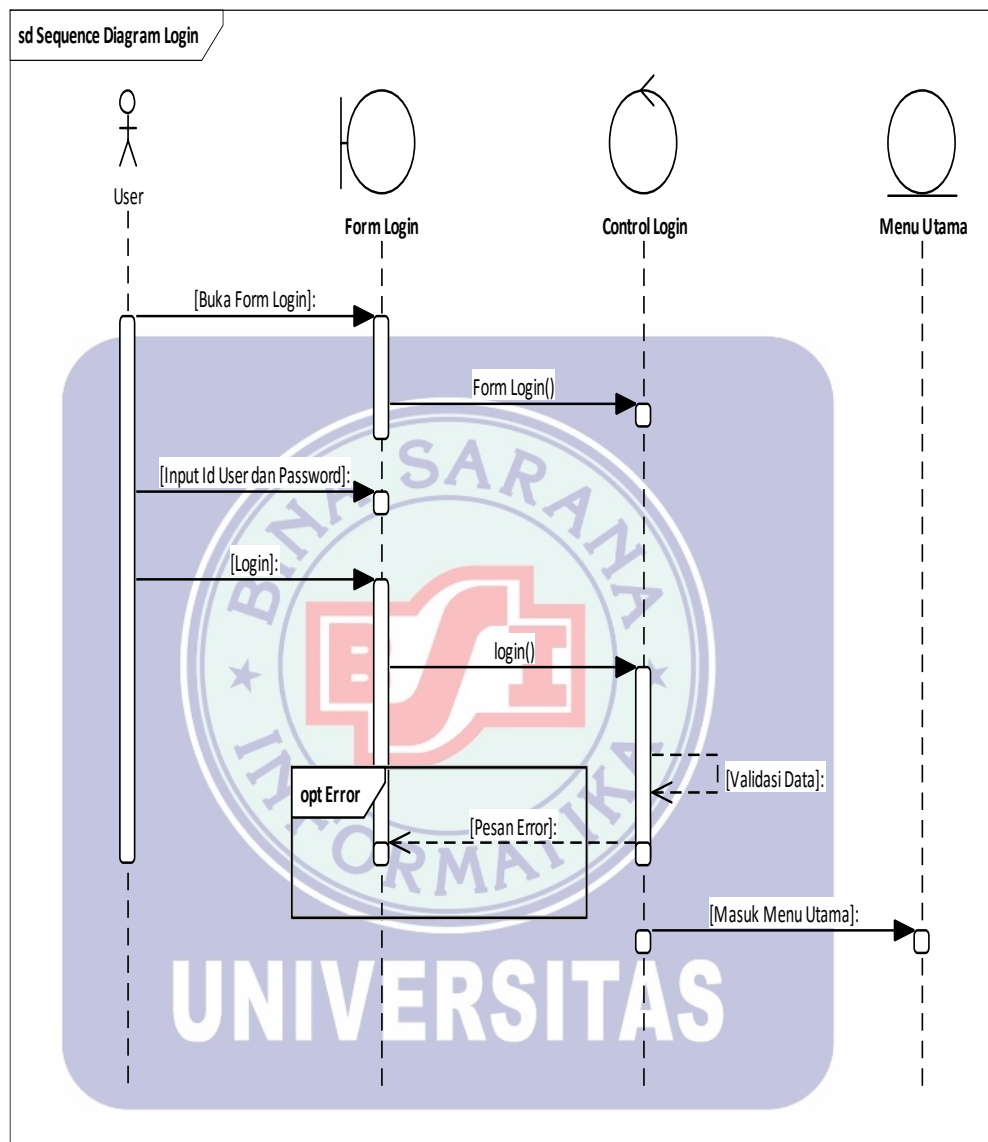


Gambar III.14

Class Diagram

3.4.5. Sequence Diagram

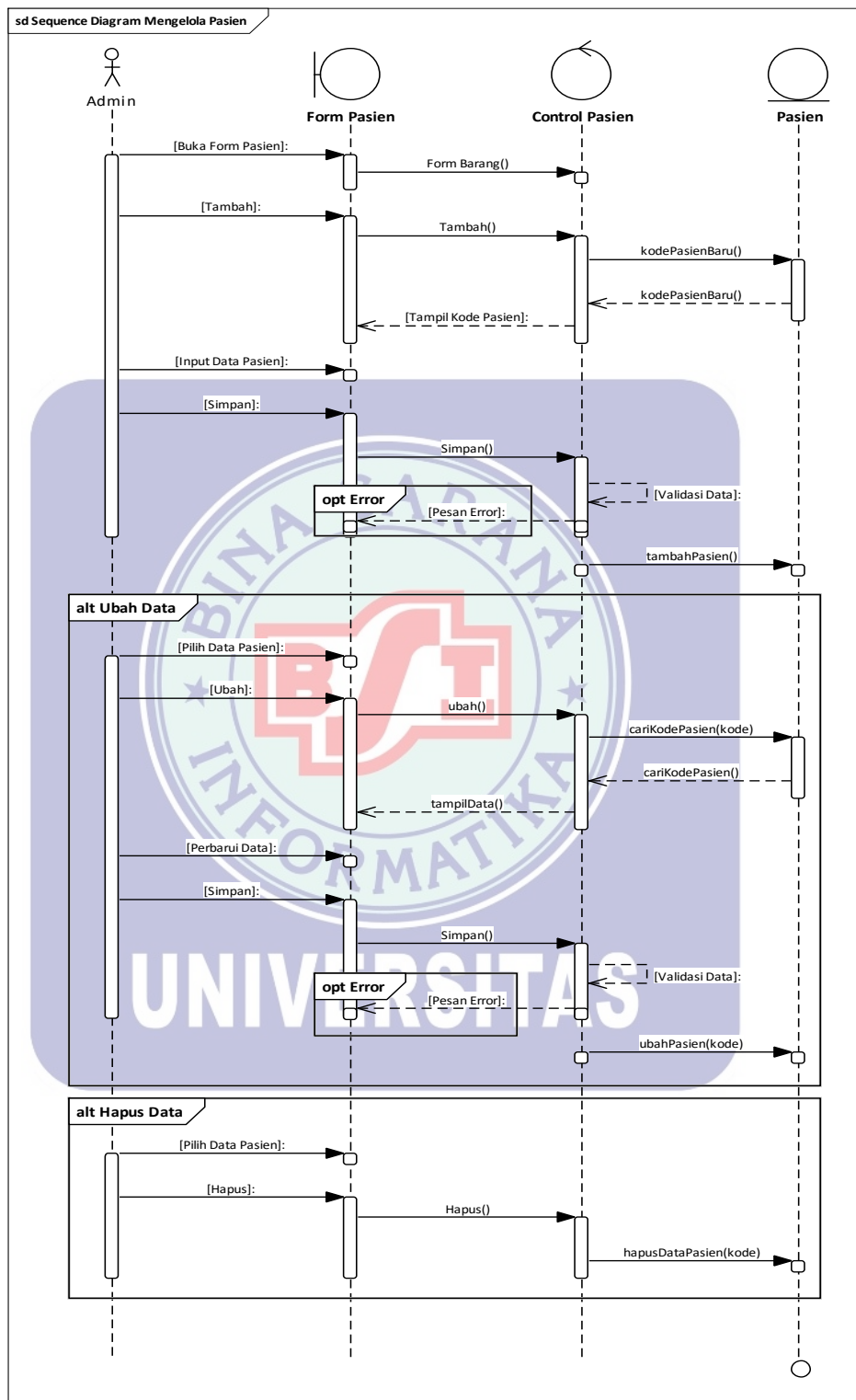
1. Sequence Diagram Login



Gambar III.15

Sequence Diagram Login

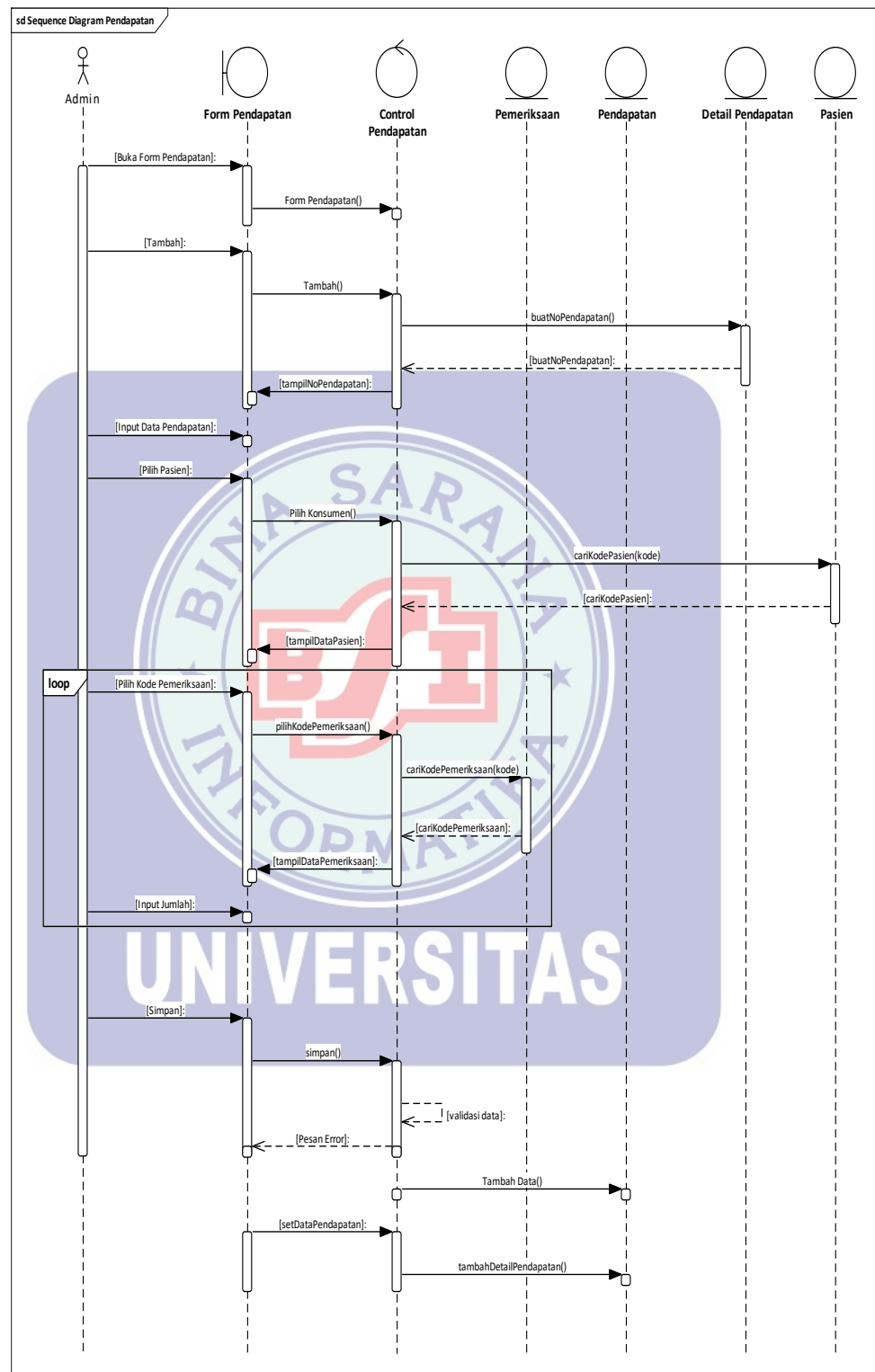
2. Sequence Diagram Mengelola Pasien



Gambar III.16

Sequence Diagram Pasien

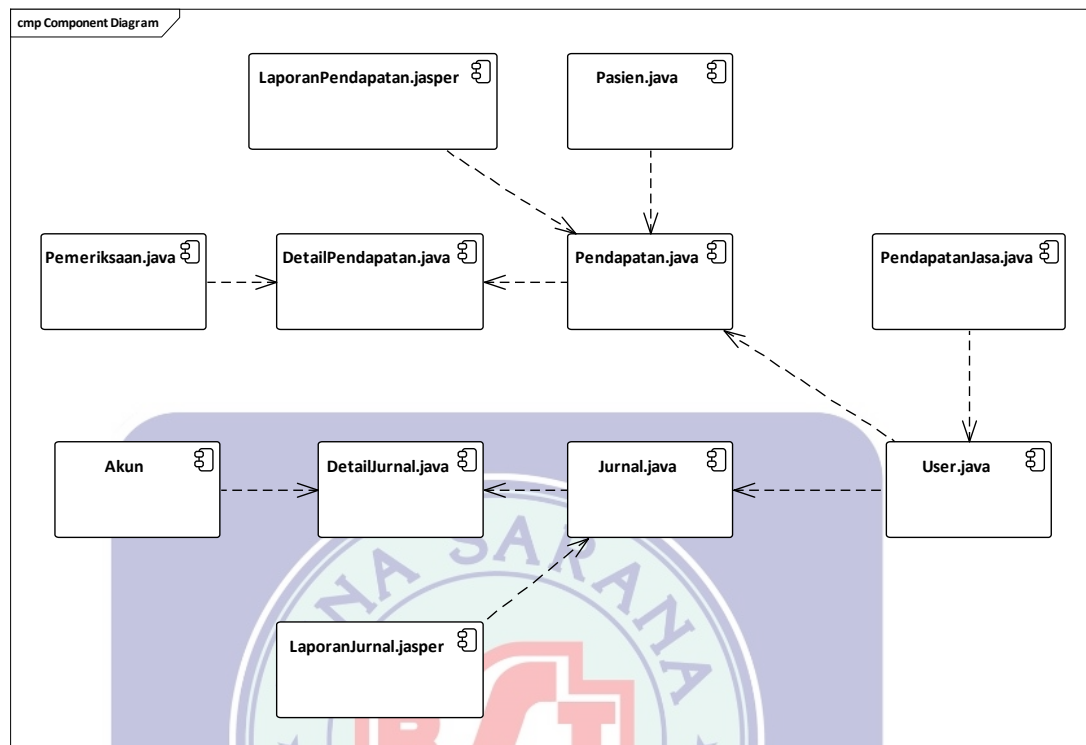
3. Sequence Diagram Pendapatan



Gambar III.17

Sequence Diagram Pendapatan

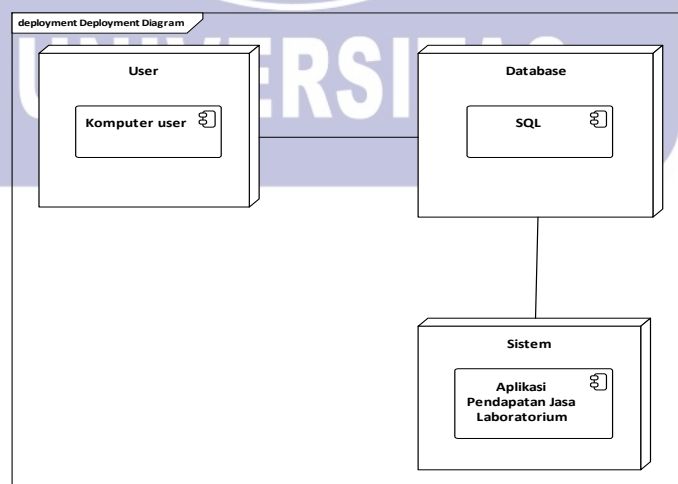
3.4.6. Component Diagram



Gambar III.18

Component Diagram

3.4.7. Deployment Diagram

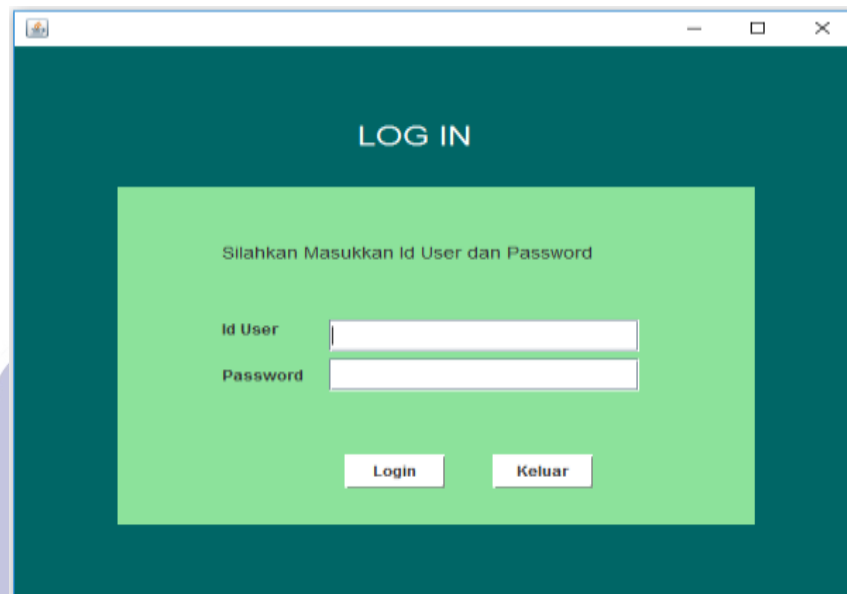


Gambar III.19

Deployment Diagram

3.4.8. User Interface

1. Tampilan Halaman *Login*



A screenshot of a web application's login page. The page has a dark teal background. At the top center, the text "LOG IN" is displayed in white. Below this, a light green rectangular box contains the text "Silahkan Masukkan Id User dan Password". Underneath, there are two input fields: "Id User" and "Password". At the bottom of the green box, there are two buttons: "Login" and "Keluar".

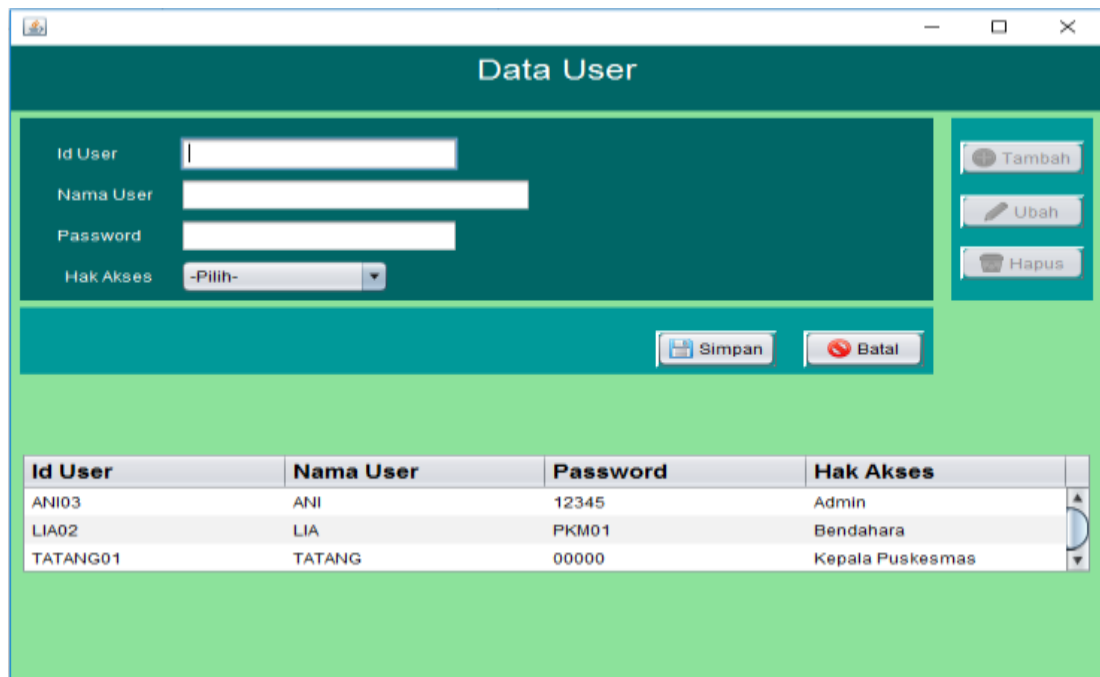
Gambar III.20
Tampilan Halaman *Login*

2. Tampilan Menu Utama



Gambar III.21
Tampilan Menu Utama

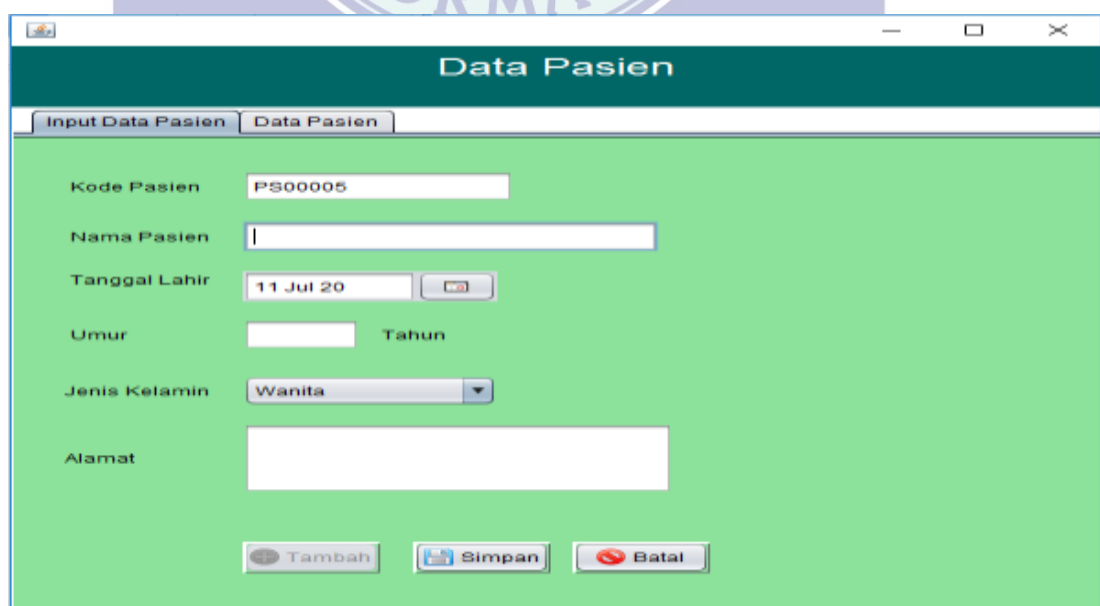
3. Tampilan Form User



Id User	Nama User	Password	Hak Akses
ANI03	ANI	12345	Admin
LIA02	LIA	PKM01	Bendahara
TATANG01	TATANG	00000	Kepala Puskesmas

Gambar III.22
Tampilan Form User

4. Tampilan Form Input Pasien



Gambar III.23
Tampilan Form Input Pasien

5. Tampilan Form Data Pasien

Kode Pasi...	Nama Pasi...	Tanggal L...	Umur	Jenis Kela...	Alamat
PS00001	ALISYA	2016-09-25	4	Wanita	JATISARI
PS00002	INA	2001-03-03	19	Wanita	BALONGGAN...
PS00003	CICI	2006-01-17	14	Wanita	SITUDAM
PS00004	TITIN	1999-07-10	21	Wanita	JATISARI

Gambar III.24
Tampilan Form Data Pasien

6. Tampilan Form Pemeriksaan Laboratorium

Kode Pemeriksaan	Nama Pemeriksaan	Harga
JS001	ASAM URAT	30000
JS002	GULA DARAH SEWAKTU	15000
JS003	KOLESTEROL TOTAL	30000
JS004	GOLONGAN DARAH	10000

Gambar III.25
Tampilan Form Pemeriksaan Laboratorium

7. Tampilan Form Akun

Kode Akun	Nama Akun	Jenis Akun	Saldo Normal
111	KAS	Harta	Debit
300	MODAL	Modal	Debit
400	PENDAPATAN USAHA	Harta	Debit

Gambar III.26

Tampilan Form Akun

8. Tampilan Form Input Pendapatan Laboratorium

Kode Pemeriksaan	Nama Pemeriksaan	Harga	Jumlah	Sub Total
------------------	------------------	-------	--------	-----------

Gambar III.27

Tampilan Form Input Pendapatan Laboratorium

9. Tampilan Form Data Pendapatan Laboratorium

No Transaksi	Tanggal Tra...	Nama Pasien	Kode Peme...	Nama Peme...	Harga	Jumlah	Sub Total
TR00001	2020-07-11	TITIN	JB004	GOLONGAN DAR...	10000	1	10000
			JB002	GULA DARAH SE...	15000	1	15000

Gambar III.28

Tampilan Form Data Pendapatan Laboratorium

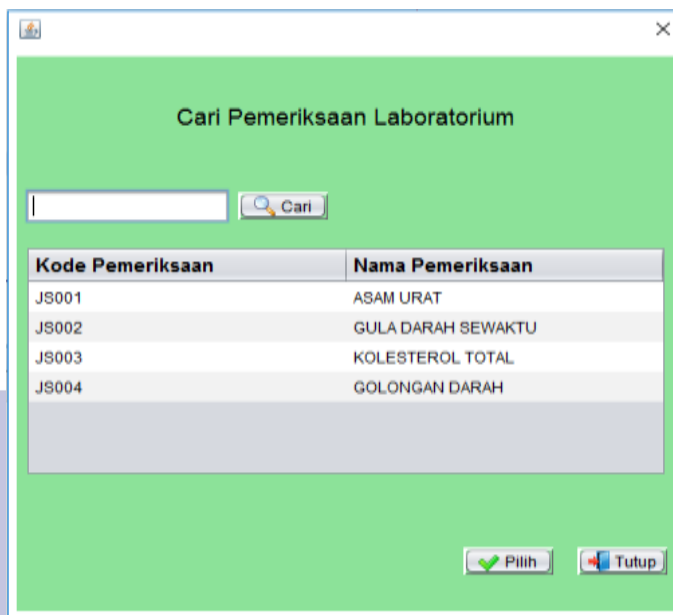
10. Tampilan Cari Pasien

Kode Pasien	Nama
PS00001	ALISYA
PS00002	INA
PS00003	CICI
PS00004	TITIN

Gambar III.29

Tampilan Cari Pasien

11. Tampilan Cari Pemeriksaan Laboratorium

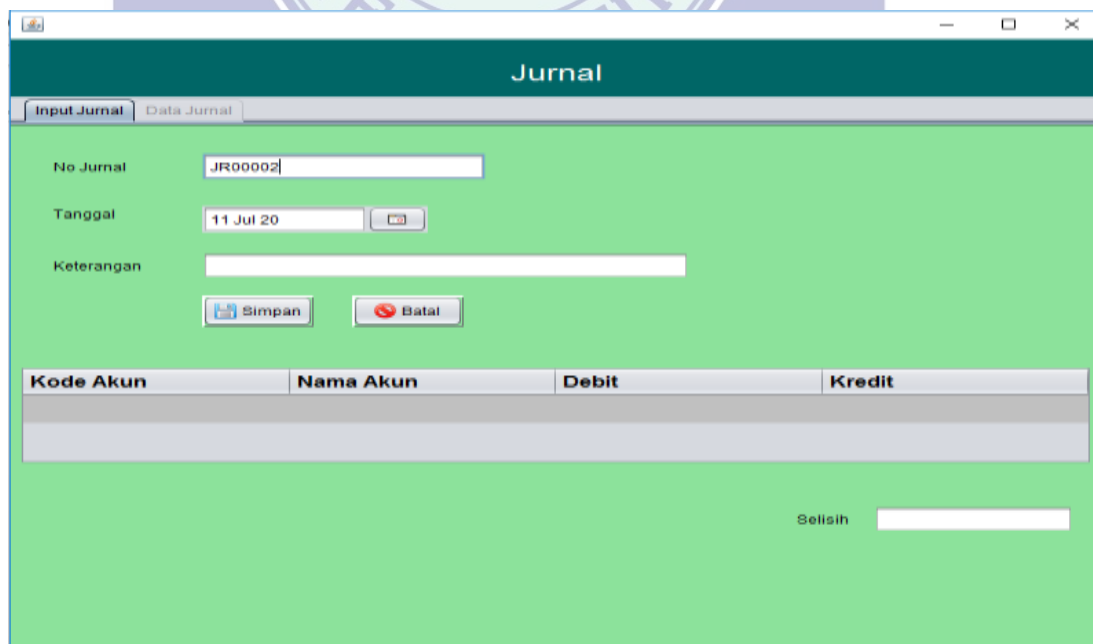


Cari Pemeriksaan Laboratorium

Kode Pemeriksaan	Nama Pemeriksaan
JS001	ASAM URAT
JS002	GULA DARAH SEWAKTU
JS003	KOLESTEROL TOTAL
JS004	GOLONGAN DARAH

Gambar III.30
Tampilan Cari Pemeriksaan Laboratorium

12. Tampilan Form Input Jurnal



Jurnal

No Jurnal:

Tanggal:

Keterangan:

Kode Akun	Nama Akun	Debit	Kredit

Selisih:

Gambar III.31
Tampilan Form Input Jurnal

13. Tampilan Form Data Jurnal

Jurnal

Input Jurnal | **Data Jurnal**

Tambah Tanggal Jurnal

No Jurnal	Tanggal Ju...	Keterangan	Kode Akun	Nama Akun	Debet	Kredit
JR00001	2020-09-07	Penerimaan Ua...	111	KAS	300000	0
			400	PENDAPATAN U...	0	300000

Gambar III.32

Tampilan Form Data Jurnal

14. Tampilan Cari Akun

Cari Akun

Kode Akun	Nama Akun
111	KAS
300	MODAL
400	PENDAPATAN USAHA

Gambar III.33

Tampilan Cari Akun

15. Tampilan Form Laporan



Gambar III.34

Tampilan Form Laporan

3.5. Implementasi

3.5.1. Code Generation

1. Model Pendapatan

```
package model;
```

```
import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
public class Pendapatan {
```

```

private Connection Koneksi;
private String no_pendapatan;
private Date tgl_pendapatan;
private String kode_pasien;
private String id_user;
private Integer total;

private String PesanError;

public Pendapatan() {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        Koneksi =
DriverManager.getConnection("jdbc:mysql://localhost/pendapatan_jasa", "root",
""");
    } catch (ClassNotFoundException | SQLException ex) {
        this.PesanError = ex.getMessage();
        ex.printStackTrace();
    }
}

public Connection getKoneksi() {
    return Koneksi;
}

public void setKoneksi(Connection Koneksi) {
    this.Koneksi = Koneksi;
}

public String getNo_pendapatan() {
    return no_pendapatan;
}

public void setNo_pendapatan(String no_pendapatan) {
    this.no_pendapatan = no_pendapatan;
}

public Date getTgl_pendapatan() {
    return tgl_pendapatan;
}

public void setTgl_pendapatan(Date tgl_pendapatan) {
    this.tgl_pendapatan = tgl_pendapatan;
}

public String getKode_pasien() {
    return kode_pasien;
}

```

```

public void setKode_pasien(String kode_pasien) {
    this.kode_pasien = kode_pasien;
}

public String getId_user() {
    return id_user;
}

public void setId_user(String id_user) {
    this.id_user = id_user;
}

public Integer getTotal() {
    return total;
}

public void setTotal(Integer total) {
    this.total = total;
}

public void setPesanError(String PesanError) {
    this.PesanError = PesanError;
}

public List<Pendapatan> daftarpendapatan() {
    try {
        this.PesanError = "";

        String Cmd = "SELECT * FROM pendapatan";
        List<Pendapatan> lp = new ArrayList();
        Statement st = Koneksi.createStatement();
        ResultSet rs = st.executeQuery(Cmd);
        while (rs.next()) {
            Pendapatan br = new Pendapatan();
            br.setNo_pendapatan(rs.getString("no_pendapatan"));
            br.setTgl_pendapatan(rs.getDate("tgl_pendapatan"));
            br.setKode_pasien(rs.getString("kode_pasien"));
            br.setId_user(rs.getString("id_user"));
            br.setTotal(rs.getInt("total"));

            lp.add(br);
        }
        return lp;
    } catch (SQLException ex) {
        Logger.getLogger(Pendapatan.class.getName()).log(Level.SEVERE, null,
ex);
        this.PesanError = ex.getMessage();
        return null;
    }
}

```

```

    }
}

public List<Pendapatan> cariTglPendapatan(Date tgl1, Date tgl2) {
    try {
        this.PesanError = "";

        String Cmd = "SELECT * FROM pendapatan WHERE tgl_pendapatan
        BETWEEN ? AND ?";
        List<Pendapatan> lp = new ArrayList();

        PreparedStatement ps = Koneksi.prepareStatement(Cmd);
        ps.setDate(1, tgl1);
        ps.setDate(2, tgl2);
        ResultSet rs = ps.executeQuery();
        while (rs.next()) {
            Pendapatan br = new Pendapatan();
            br.setNo_pendapatan(rs.getString("no_pendapatan"));
            br.setTgl_pendapatan(rs.getDate("tgl_pendapatan"));
            br.setKode_pasien(rs.getString("kode_pasien"));
            br.setId_user(rs.getString("id_user"));
            br.setTotal(rs.getInt("total"));

            lp.add(br);
        }
        return lp;
    } catch (SQLException ex) {
        Logger.getLogger(Pendapatan.class.getName()).log(Level.SEVERE, null,
ex);
        this.PesanError = ex.getMessage();
        return null;
    }
}

public int tambahPendapatan() {
    try {
        this.PesanError = "";

        String Cmd = "INSERT INTO pendapatan (no_pendapatan,
        tgl_pendapatan, kode_pasien, id_user, total) VALUES (?, ?, ?, ?, ?)";
        PreparedStatement ps = Koneksi.prepareStatement(Cmd);

        ps.setString(1, this.no_pendapatan);
        ps.setDate(2, this.tgl_pendapatan);
        ps.setString(3, this.kode_pasien);
        ps.setString(4, this.id_user);
        ps.setInt(5, this.total);
    }
}

```

```

        return ps.executeUpdate();
    } catch (SQLException ex) {
        System.err.println(ex.getMessage());
        this.PesanError = ex.getMessage();
        return -1;
    }
}

public String noAuto() {
    try {
        String Cmd = "SELECT * FROM pendapatan ORDER BY
SUBSTR(no_pendapatan,3,7) * 1 DESC LIMIT 1";
        Statement st = Koneksi.createStatement();
        ResultSet rs = st.executeQuery(Cmd);
        if (rs.next()) {
            String No = rs.getString("no_pendapatan");
            No = No.substring(2, 7);
            int urut = Integer.parseInt(No) + 1;
            No = String.valueOf(urut);
            No = "TR" + String.format("%05d", Integer.parseInt(No));
            return No;
        } else {
            return "TR00001";
        }
    } catch (SQLException ex) {
        this.PesanError = ex.getMessage();
        return "";
    }
}
}

```

2. Form Pendapatan

```

package form;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.Toolkit;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.text.DateFormat;
import java.text.DecimalFormat;
import java.text.NumberFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
import java.util.ListIterator;

```

```

import javax.swing.DefaultCellEditor;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumn;
import model.DetailPendapatan;
import model.Pemeriksaan;
import model.Pasien;
import pendapatanjasa.Pendapatanjasa;
import model.Pendapatan;

public class FormPendapatan extends javax.swing.JFrame {

    String NoPendapatan = "";
    String Status = "";
    int total = 0;

    int row = -1;

    Pemeriksaan pemeriksaan = new Pemeriksaan();
    Pendapatan pendapatan = new Pendapatan();
    DetailPendapatan detailPendapatan = new DetailPendapatan();
    Pasien pasien = new Pasien();

    DefaultComboBoxModel tml = null;
    DefaultTableModel model, modelData;

    TableColumn cKode, cJumlah;

    JTextField textKode = new JTextField();
    JTextField textJumlah = new JTextField();

    NumberFormat nf = new DecimalFormat("#");

    /**
     * Creates new form FormTransaksi
     */
    public FormPendapatan() {
        initComponents();
        Toolkit tk = Toolkit.getDefaultToolkit();
        Dimension d = tk.getScreenSize();
        int x, y;
        x = (int) ((d.getWidth() - getSize().width) / 2);
        y = (int) ((d.getHeight() - getSize().height) / 2);
        this.getContentPane().setBackground(Color.WHITE);

        setLocation(x, y);
    }

```

```

model = (DefaultTableModel) tblDetail.getModel();
modelData = (DefaultTableModel) tblData.getModel();

textKode.setFont(new Font("BellGothic BT", 0, 16));
textJumlah.setFont(new Font("BellGothic BT", 0, 16));

tblDetail.getTableHeader().setFont(new Font("BellGothic", 1, 16));
tblDetail.setRowHeight(30);
tblData.getTableHeader().setFont(new Font("BellGothic", 1, 16));
tblData.setRowHeight(30);

cKode = (TableColumn) tblDetail.getColumnModel().getColumn(0);
cJumlah = (TableColumn) tblDetail.getColumnModel().getColumn(3);

cKode.setCellEditor(new DefaultCellEditor(textKode));
cJumlah.setCellEditor(new DefaultCellEditor(textJumlah));

textKode.addKeyListener(new KeyAdapter() {
    public void keyPressed(KeyEvent e) {
        if (e.getKeyCode() == KeyEvent.VK_ENTER) {
            row = tblDetail.getSelectedRow();
            textKode.setText(textKode.getText().toUpperCase());

            if (!"".equals(textKode.getText())) {
                getDataPemeriksaan(textKode.getText(), row);

                for (int i = 0; i < model.getRowCount() - 1; i++) {
                    String kode = model.getValueAt(i, 0).toString();
                    if (kode.equalsIgnoreCase(textKode.getText()) && row != i) {
                        model.setValueAt(null, row, 0);
                        model.setValueAt(null, row, 1);
                        model.setValueAt(null, row, 2);
                        model.setValueAt(null, row, 3);
                        model.setValueAt(null, row, 4);
                        textKode.setText(null);
                        JOptionPane.showMessageDialog(rootPane, "Data Sudah
Ada!", "info", JOptionPane.ERROR_MESSAGE);
                        break;
                    }
                }
            }
        }
        if (e.getKeyCode() == KeyEvent.VK_F1) {
            textKode.setText(new FormCariPemeriksaan(null,
true).cariPemeriksaan());
            row = tblDetail.getSelectedRow();
            textKode.setText(textKode.getText().toUpperCase());
            if (!"".equals(textKode.getText())) {

```

```

        getDataPemeriksaan(textKode.getText(), row);

        for (int i = 0; i < model.getRowCount() - 1; i++) {
            String kode = model.getValueAt(i, 0).toString();
            if (kode.equalsIgnoreCase(textKode.getText()) && row != i) {
                model.setValueAt(null, row, 0);
                model.setValueAt(null, row, 1);
                model.setValueAt(null, row, 2);
                model.setValueAt(null, row, 3);
                model.setValueAt(null, row, 4);
                textKode.setText(null);
                JOptionPane.showMessageDialog(rootPane, "Data Sudah
Ada!", "Info", JOptionPane.ERROR_MESSAGE);
                break;
            }
        }
    }
});

textJumlah.addKeyListener(new KeyAdapter() {
    public void keyTyped(KeyEvent e) {
        if (!Character.isDigit(e.getKeyChar()) && e.getKeyChar() !=
KeyEvent.VK_BACK_SPACE && e.getKeyChar() != KeyEvent.VK_DELETE)
        {
            textJumlah.setEditable(false);
            JOptionPane.showMessageDialog(rootPane, "Input Hanya Boleh
Berupa Angka!", "Error", JOptionPane.ERROR_MESSAGE);
        } else {
            textJumlah.setEditable(true);
            try {
                textJumlah.setText(String.format("%d",
Integer.parseInt(textJumlah.getText())));
            } catch (Exception ex) {
            }
        }
        if (textJumlah.getText().equals("")) {
            textJumlah.setText("0");
        }
    }

    public void keyPressed(KeyEvent e) {
        if (e.getKeyCode() == KeyEvent.VK_ENTER || e.getKeyCode() ==
KeyEvent.VK_TAB) {
            try {
                row = tblDetail.getSelectedRow();

```

```

        int sub_total = Integer.parseInt(textJumlah.getText()) *
Integer.parseInt(model.getValueAt(row, 2).toString());
        model.setValueAt(sub_total, row, 4);
        hitungTotal();

        if (textJumlah.getText().equals("")) {
            textJumlah.setText("1");
        }
    } catch (Exception ex) {
    }
}
});

tblDetail.setSelectionBackground(Color.lightGray);

aktif();

java.util.Date dt = new java.util.Date();
dtanggal.setDate(dt);

tabs.setSelectedIndex(1);

daftarPendapatan();
this.NoPendapatan = "";
aktif();
}

private void aktif() {
    bbatal.setEnabled(!"".equals(Status));
    bsimpan.setEnabled(!"".equals(Status));
    btambah.setEnabled(!"".equals(Status));

    tabs.setEnabledAt(0, !"".equals(Status));
    tabs.setEnabledAt(1, !"".equals(Status));
}

private void bersihForm() {
    tpasien.setText("");
    tno.setText("");
    model.setRowCount(0);
    tblDetail.repaint();
    tttotal.setText("");
}

private void getDataPemeriksaan(String kode, int row) {
    pemeriksaan = pemeriksaan.cariKodePemeriksaan(kode);
    if (pemeriksaan.getNama_pemeriksaan() == null) {

```

```
JOptionPane.showMessageDialog(rootPane, "Data Pemeriksaan Yang  
Anda Cari Tidak Ada!", "Error", JOptionPane.ERROR_MESSAGE);
```

```
model.setValueAt(null, row, 0);  
model.setValueAt(null, row, 1);  
model.setValueAt(null, row, 2);  
model.setValueAt(null, row, 3);  
model.setValueAt(null, row, 4);
```

```
} else {  
    model.setValueAt(kode.toUpperCase(), row, 0);  
    model.setValueAt(pemeriksaan.getNama_pemeriksaan(), row, 1);  
    model.setValueAt(pemeriksaan.getHarga(), row, 2);
```

```
}  
}
```

```
private void daftarPendapatan() {  
    List<Pendapatan> lp = pendapatan.daftarpendapatan();  
    ListIterator li = lp.listIterator();  
    modelData.setRowCount(0);  
    tbldata.repaint();  
    String No = "";  
    Object[] data = null;  
    while (li.hasNext()) {  
        pendapatan = (Pendapatan) li.next();  
        pasien = pasien.cariKodePasien(pendapatan.getKode_pasien());  
        List<DetailPendapatan> ld =  
        detailPendapatan.daftardetailpendapatan(pendapatan.getNo_pendapatan());  
        ListIterator<DetailPendapatan> it = ld.listIterator();  
        while (it.hasNext()) {  
            detailPendapatan = it.next();  
            pemeriksaan =  
            pemeriksaan.cariKodePemeriksaan(detailPendapatan.getKode_pemeriksaan());  
            if (!No.equalsIgnoreCase(pendapatan.getNo_pendapatan())) {  
                Object[] jdata = {  
                    pendapatan.getNo_pendapatan(),  
                    pendapatan.getTgl_pendapatan(),  
                    pasien.getNama_pasien(),  
                    detailPendapatan.getKode_pemeriksaan(),  
                    pemeriksaan.getNama_pemeriksaan(),  
                    pemeriksaan.getHarga(),  
                    detailPendapatan.getJumlah(),  
                    detailPendapatan.getSub_total()  
                };  
                data = jdata;  
            } else {  
                Object[] jdata = {  
                    "",
```

```

        """,
        """,
        detailPendapatan.getKode_pemeriksaan(),
        pemeriksaan.getNama_pemeriksaan(),
        pemeriksaan.getHarga(),
        detailPendapatan.getJumlah(),
        detailPendapatan.getSub_total()
    };
    data = jdata;
}
modelData.addRow(data);
No = pendapatan.getNo_pendapatan();
}
}

private void daftarPendapatan(Date tgl1, Date tgl2) {
    List<Pendapatan> lp = pendapatan.cariTglPendapatan(new
    java.sql.Date(tgl1.getTime()), new java.sql.Date(tgl2.getTime()));
    ListIterator li = lp.listIterator();
    modelData.setRowCount(0);
    tbldata.repaint();
    String No = "";
    Object[] data = null;
    while (li.hasNext()) {
        pendapatan = (Pendapatan) li.next();
        pasien = pasien.cariKodePasien(pendapatan.getKode_pasien());
        List<DetailPendapatan> ld =
        detailPendapatan.daftardetailpendapatan(pendapatan.getNo_pendapatan());
        ListIterator<DetailPendapatan> it = ld.listIterator();
        while (it.hasNext()) {
            detailPendapatan = it.next();
            pemeriksaan =
            pemeriksaan.cariKodePemeriksaan(detailPendapatan.getKode_pemeriksaan());
            if (!No.equalsIgnoreCase(pendapatan.getNo_pendapatan())) {
                Object[] jdata = {
                    pendapatan.getNo_pendapatan(),
                    pendapatan.getTgl_pendapatan(),
                    pasien.getNama_pasien(),
                    detailPendapatan.getKode_pemeriksaan(),
                    pemeriksaan.getNama_pemeriksaan(),
                    pemeriksaan.getHarga(),
                    detailPendapatan.getJumlah(),
                    detailPendapatan.getSub_total()
                };

                data = jdata;
            } else {
                Object[] jdata = {

```

```

        """,
        """,
        """,
        detailPendapatan.getKode_pemeriksaan(),
        pemeriksaan.getNama_pemeriksaan(),
        pemeriksaan.getHarga(),
        detailPendapatan.getJumlah(),
        detailPendapatan.getSub_total()
    };
    data = jdata;
}
modelData.addRow(data);
No = pendapatan.getNo_pendapatan();
}
}

private void hitungTotal() {
    this.total = 0;
    for (int i = 0; i < model.getRowCount(); i++) {
        try {
            this.total += Integer.parseInt(model.getValueAt(i, 4).toString());
        } catch (Exception ex) {
        }
    }
    tttotal.setText(nf.format(this.total));
}

// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jDateChooser2 = new com.toedter.calendar.JDateChooser();
    tabs = new javax.swing.JTabbedPane();
    jPanel1 = new javax.swing.JPanel();
    tno_transaksi = new javax.swing.JLabel();
    ttanggal = new javax.swing.JLabel();
    tno = new javax.swing.JTextField();
    bsimpan = new javax.swing.JButton();
    bbatal = new javax.swing.JButton();
    jlabel = new javax.swing.JLabel();
    tpasien = new javax.swing.JTextField();
    jScrollPane3 = new javax.swing.JScrollPane();
    tblDetail = new javax.swing.JTable();
    tnama_pasien = new javax.swing.JTextField();
    tttotal = new javax.swing.JTextField();
    jLabel3 = new javax.swing.JLabel();
    dtanggal = new com.toedter.calendar.JDateChooser();

```

```

jPanel2 = new javax.swing.JPanel();
jScrollPane2 = new javax.swing.JScrollPane();
tbldata = new javax.swing.JTable();
jLabel5 = new javax.swing.JLabel();
bcari = new javax.swing.JButton();
btambah = new javax.swing.JButton();
tgl1 = new com.toedter.calendar.JDateChooser();
tgl2 = new com.toedter.calendar.JDateChooser();
jPanel3 = new javax.swing.JPanel();
jLabel6 = new javax.swing.JLabel();

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"
    }
));
jScrollPane1.setViewportView(jTable1);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

tabs.setBackground(new java.awt.Color(255, 255, 255));

jPanel1.setBackground(new java.awt.Color(140, 226, 155));

tno_transaksi.setText("No Pendapatan");
ttanggal.setText("Tanggal");

bsimpan.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/gambar/icon/simpan.png"))); //
NOI18N
bsimpan.setText("Simpan");
bsimpan.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED))
;

bsimpan.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        bsimpanActionPerformed(evt);
    }
});

```

```

        bbatal.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/gambar/icon/batal.png"))); //
NOI18N
        bbatal.setText("Batal");
        bbatal.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED))
;
        bbatal.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bbatalActionPerformed(evt);
            }
        });

jlabel.setText("Kode Pasien");

tpasien.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tpasienActionPerformed(evt);
    }
});
tpasien.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyPressed(java.awt.event.KeyEvent evt) {
        tpasienKeyPressed(evt);
    }
});

tbldetail.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null}
    },
    new String [] {
        "Kode Pemeriksaan", "Nama Pemeriksaan", "Harga", "Jumlah", "Sub
Total"
    }
) {
    boolean[] canEdit = new boolean [] {
        true, false, false, true, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {

```

```

        return canEdit [columnIndex];
    }
});
tblDetail.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblDetailMouseClicked(evt);
    }
});
tblDetail.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyPressed(java.awt.event.KeyEvent evt) {
        tblDetailKeyPressed(evt);
    }
});
jScrollPane3.setViewportViewView(tblDetail);
if (tblDetail.getColumnModel().getColumnCount() > 0) {
    tblDetail.getColumnModel().getColumn(0).setResizable(false);
    tblDetail.getColumnModel().getColumn(1).setResizable(false);
    tblDetail.getColumnModel().getColumn(2).setResizable(false);
    tblDetail.getColumnModel().getColumn(3).setResizable(false);
    tblDetail.getColumnModel().getColumn(4).setResizable(false);
}

jLabel3.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N
jLabel3.setText("Jumlah Keseluruhan");

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(42, 42, 42)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
            .addComponent(tanggal,
                javax.swing.GroupLayout.Alignment.LEADING,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jLabel,
                javax.swing.GroupLayout.Alignment.LEADING,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(tno_transaksi,
                javax.swing.GroupLayout.Alignment.LEADING))
        .addGap(35, 35, 35)

```

```

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup())
    .addGap(35, 35, 35)
    .addComponent(bsimpan,
javax.swing.GroupLayout.PREFERRED_SIZE, 77,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addComponent(tpasien,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 132,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(bbatal)
    .addComponent(tnama_pasien,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addComponent(dtanggal,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(tno, javax.swing.GroupLayout.DEFAULT_SIZE,
154, Short.MAX_VALUE))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
    .addContainerGap(645, Short.MAX_VALUE)
    .addComponent(jLabel3)
    .addGap(18, 18, 18)
    .addComponent(tttotal, javax.swing.GroupLayout.PREFERRED_SIZE,
139, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(19, 19, 19))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
    .addContainerGap()
    .addComponent(jScrollPane3)
    .addContainerGap())
);

```

```

jPanel1Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new
java.awt.Component[] {bbatal, bsimpan});

jPanel1Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new
java.awt.Component[] {dtanggal, tnama_pasien, tno, tpasien});

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(16, 16, 16)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(tno_transaksi)
    .addComponent(tno, javax.swing.GroupLayout.PREFERRED_SIZE,
20, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(ttanggal)
    .addComponent(dtanggal,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(27, 27, 27)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(tpasien,
javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(tnama_pasien,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addComponent(jlabel))
    .addGap(50, 50, 50)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(bsimpan,
javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(bbatal))
        .addGap(26, 26, 26)
        .addComponent(jScrollPane3,
javax.swing.GroupLayout.DEFAULT_SIZE, 125, Short.MAX_VALUE)
        .addGap(18, 18, 18)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(tttotal,
javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel3))
        .addGap(89, 89, 89))
    );

    jPanel1Layout.linkSize(javax.swing.SwingConstants.VERTICAL, new
java.awt.Component[] {dtanggal, tnama_pasien, tno, tpasien});

    jPanel1Layout.linkSize(javax.swing.SwingConstants.VERTICAL, new
java.awt.Component[] {bbatal, bsimpan});

    tabs.addTab("Input Transaksi", jPanel1);

    jPanel2.setBackground(new java.awt.Color(140, 226, 155));

    tbldata.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null}
        },
        new String [] {
            "No Transaksi", "Tanggal Transaksi", "Nama Pasien", "Kode
Pemeriksaan", "Nama Pemeriksaan", "Harga", "Jumlah", "Sub Total"
        }
    ) {
        boolean[] canEdit = new boolean [] {

```

```

        false, false, false, false, false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
tbldata.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tbldataMouseClicked(evt);
    }
});
jScrollPane2.setViewportViewView(tbldata);

jLabel5.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
jLabel5.setText("Tanggal Transaksi");

bcari.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/gambar/icon/cari.png"))); //
NOI18N
bcari.setText("Cari");
bcari.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED))
;
bcari.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        bcariActionPerformed(evt);
    }
});

btambah.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/gambar/icon/tambah.png"))); //
NOI18N
btambah.setText("Tambah");
btambah.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED))
;
btambah.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btambahActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(

```

```

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane2,
javax.swing.GroupLayout.DEFAULT_SIZE, 916, Short.MAX_VALUE))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
                .addGap(22, 22, 22)
                .addComponent(btambah,
javax.swing.GroupLayout.PREFERRED_SIZE, 77,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel5)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED
                )
                .addComponent(tgl1,
javax.swing.GroupLayout.PREFERRED_SIZE, 129,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(tgl2,
javax.swing.GroupLayout.PREFERRED_SIZE, 115,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(22, 22, 22)
                .addComponent(bcari)))
            .addContainerGap())
    );

jPanel2Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new
java.awt.Component[] {bcari, btambah});

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(46, 46, 46)

```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
        .addComponent(btambah,
            javax.swing.GroupLayout.PREFERRED_SIZE, 32,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel5))
        .addComponent(bcari,
            javax.swing.GroupLayout.Alignment.LEADING,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addComponent(tgl2,
            javax.swing.GroupLayout.PREFERRED_SIZE, 32,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tgl1,
            javax.swing.GroupLayout.PREFERRED_SIZE, 32,
            javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(34, 34, 34)
        .addComponent(jScrollPane2,
            javax.swing.GroupLayout.DEFAULT_SIZE, 266, Short.MAX_VALUE)
        .addGap(149, 149, 149))
    );
```

```
jPanel2Layout.linkSize(javax.swing.SwingConstants.VERTICAL, new
java.awt.Component[] {bcari, btambah, tgl1, tgl2});
```

```
tabs.addTab("Data Transaksi", jPanel2);
```

```
jPanel3.setBackground(new java.awt.Color(0, 102, 102));
```

```
jLabel6.setFont(new java.awt.Font("Arial", 0, 24)); // NOI18N
```

```
jLabel6.setForeground(new java.awt.Color(255, 255, 255));
```

```
jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
```

```
jLabel6.setText("Pendapatan Laboratorium");
```

```
        javax.swing.GroupLayout jPanel3Layout = new
        javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(
```

```
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```

        .addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGap(40, 40, 40))
    );

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(tabs)
    .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addComponent(jPanel3,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tabs))
    );

    pack();
} // </editor-fold>

private void bbatalActionPerformed(java.awt.event.ActionEvent evt) {
    Status = "";
    tno.setText("");

    DateFormat df = new SimpleDateFormat("dd/MM/yyyy");
    java.util.Date dt = new java.util.Date();
    dtanggal.setDateFormatString(df.format(dt));
    tabs.setSelectedIndex(1);
    bersihForm();
    aktif();
}

```

```

    }

    private void tblDetailMouseClicked(java.awt.event.MouseEvent evt) {
        try {
            row = tblDetail.getSelectedRow();
            NoPendapatan = (String) model.getValueAt(row, 0);
        } catch (Exception e) {
        }
    }

    private void tblDetailKeyPressed(java.awt.event.KeyEvent evt) {
        if (evt.getKeyCode() == KeyEvent.VK_DELETE) {
            if (!"".equals(NoPendapatan)) {
                row = tblDetail.getSelectedRow();
                model.removeRow(row);
                tblDetail.repaint();
                if (model.getRowCount() == 0) {
                    model.setRowCount(1);
                }
            }
        }
        if (evt.getKeyCode() == KeyEvent.VK_TAB) {
            try {
                double subtotal =
                Double.parseDouble(model.getValueAt(model.getRowCount() - 1, 4).toString());
                if (subtotal > 0) {
                    model.setRowCount(model.getRowCount() + 1);
                    tblDetail.repaint();
                }
            } catch (Exception ex) {
            }
        }
    }

    private void btambahActionPerformed(java.awt.event.ActionEvent evt) {
        this.Status = "Tambah";

        bersihForm();
        tno.setText(pendapatan.noAuto());
        aktif();
        tabs.setSelectedIndex(0);
        model.setRowCount(1);
        this.NoPendapatan = "";
    }

    private void bsimpanActionPerformed(java.awt.event.ActionEvent evt) {
        if (!"".equals(tpasien.getText())) {
            JOptionPane.showMessageDialog(rootPane, "Pasien Belum Dipilih",
            "Galat", JOptionPane.ERROR_MESSAGE);
            return;
        }
    }

```

```

    }
    if (model.getRowCount() == 1) {
        try {
            String Kode = model.getValueAt(0, 0).toString();
        } catch (Exception e) {
            JOptionPane.showMessageDialog(rootPane, "Belum Ada Pemeriksaan
Laboratorium Yang Dipilih", "Galat", JOptionPane.ERROR_MESSAGE);
            return;
        }
    }
    pendapatan.setNo_pendapatan(tno.getText());
    java.util.Date dt = new java.util.Date();
    java.sql.Date sdate = new java.sql.Date(dtanggal.getDate().getTime());
    pendapatan.setTgl_pendapatan(sdate);
    pendapatan.setKode_pasien(tpasien.getText());
    pendapatan.setTotal(Integer.parseInt(tttotal.getText()));
    pendapatan.setId_user(Pendapatanjasa.id_user);

    pendapatan.tambahPendapatan();

    for (int i = 0; i < model.getRowCount(); i++) {
        String Kode = (String) model.getValueAt(i, 0);
        int Jumlah = Integer.parseInt(model.getValueAt(i, 3).toString());
        int Harga = Integer.parseInt(model.getValueAt(i, 2).toString());
        int Subtotal = Integer.parseInt(model.getValueAt(i, 2).toString()) *
Integer.parseInt(model.getValueAt(i, 3).toString());

        detailPendapatan.setNo_pendapatan(tno.getText());
        detailPendapatan.setKode_pemeriksaan(Kode);
        detailPendapatan.setJumlah(Jumlah);
        detailPendapatan.setHarga(Harga);
        detailPendapatan.setSub_total(Subtotal);

        detailPendapatan.tambahDetailPendapatan();
    }
    tno.setText("");
    tpasien.setText("");
    model.getDataVector().removeAllElements();
    tblDetail.repaint();
    JOptionPane.showMessageDialog(rootPane, "Penerimaan Pendapatan
Berhasil", "Info", JOptionPane.INFORMATION_MESSAGE);
    tabs.setSelectedIndex(1);

    Status = "";
    daftarPendapatan();
    aktif();
}

```

```

private void bcariActionPerformed(java.awt.event.ActionEvent evt) {
    if (tgl1.getDate() == null || tgl2.getDate() == null) {
        daftarPendapatan();
    } else {
        daftarPendapatan(tgl1.getDate(), tgl2.getDate());
    }
}

private void tbldataMouseClicked(java.awt.event.MouseEvent evt) {
    int row = this.tbldata.getSelectedRow();
    this.NoPendapatan = modelData.getValueAt(row, 0).toString();
    aktif();
}

private void tpasienKeyPressed(java.awt.event.KeyEvent evt) {
    if (evt.getKeyCode() == KeyEvent.VK_F1) {
        String x = new FormCariPasien(null, true).cariPasien();
        tpasien.setText(x);
        if (x != null) {
            pasien = pasien.cariKodePasien(x);
            jLabel.setText(x);
            tnama_pasien.setText(pasien.getNama_pasien());
        } else {
            jLabel.setText("");
            tnama_pasien.setText("");
        }
        //textKode.setText(new FormCariPasien(null, true).cariPasien());
        //textKode.setText(textKode.getText().toUpperCase());
    }
}

private void tpasienActionPerformed(java.awt.event.ActionEvent evt) {
    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(FormPendapatan.class
            .getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

```

```

        java.util.logging.Logger.getLogger(FormPendapatan.class
            .getName()).log(java.util.logging.Level.SEVERE, null, ex);

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

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(FormPendapatan.class
            .getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new FormPendapatan().setVisible(true);
        }
    });
}

private javax.swing.JButton bbatal;
private javax.swing.JButton bcari;
private javax.swing.JButton bsimpan;
private javax.swing.JButton btambah;
private com.toedter.calendar.JDateChooser dtanggal;
private com.toedter.calendar.JDateChooser jDateChooser2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JTable jTable1;
private javax.swing.JLabel jLabel;
private javax.swing.JTabbedPane tabs;
private javax.swing.JTable tbldata;
private javax.swing.JTable tbldetail;
private com.toedter.calendar.JDateChooser tgl1;
private com.toedter.calendar.JDateChooser tgl2;
private javax.swing.JTextField tnama_pasien;
private javax.swing.JTextField tno;
private javax.swing.JLabel tno_transaksi;
private javax.swing.JTextField tpasien;
private javax.swing.JLabel ttanggal;
private javax.swing.JTextField tttotal;
}

```

3.5.2. Blackbox Testing

1. Blackbox Testing Halaman Login

Tabel III.24

Blackbox Testing Form Login

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Id User dan password tidak diisi kemudian klik Login	Id User (kosong) Password (kosong)	Sistem akan menolak akses dan menampilkan pesan "Id User atau Password tidak boleh kosong!"	Sesuai Harapan	Valid
2	Id User diisi sedangkan Password tidak diisi kemudian klik Login	Id User (PKMJATISAR I01) Password (kosong)	Sistem akan menolak akses dan menampilkan pesan "Id User atau Password tidak boleh kosong! "	Sesuai Harapan	Valid
3	Id User tidak di isi dan Password diisi kemudian klik Login	Id User (Kosong) Password (00000)	Sistem akan menolak akses dan menampilkan pesan "Id User atau Password tidak boleh kosong!"	Sesuai Harapan	Valid
4	Mengetikkan salah satu kondisi salah pada Id User atau password kemudian klik Login	Id User PKMJATISARI 01 (Benar) Password 11111(Salah)	Sistem akan menolak akses dan menampilkan pesan "Id User atau Password yang anda masukkan keliru"	Sesuai Harapan	Valid
5	Mengetikkan Id User dan password dengan data yang benar kemudian klik Login	Id User PKMJATISARI 01 (Benar) Password 00000(Benar)	Sistem akan menerima akses Login dan menampilkan Form Menu Utama	Sesuai Harapan	Valid

2. *Blackbox Testing* Form Akun

Tabel III.25
Blackbox Testing Form Akun

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pegujian	Kesimpulan
1	Semua <i>field</i> tidak diisi kemudian klik Simpan	Kode Akun (kosong) Nama Akun (kosong) Jenis Akun (tidak dipilih) Saldo Normal (tidak dipilih)	Sistem akan menolak akses dan menampilkan pesan "Mohon isi data dulu!"	Sesuai Harapan	<i>Valid</i>
2	Semua <i>field</i> diisi kecuali salah satu <i>field</i> kosong kemudian klik Simpan	Kode Akun (kosong) Nama Akun (pendapatan usaha) Jenis Akun (harta) Saldo Normal (debit)	Sistem akan menolak akses dan menampilkan pesan "Mohon isi data dulu!"	Sesuai Harapan	<i>Valid</i>
3	Semua <i>field</i> diisi, tetapi Nama <i>User</i> tidak diisi dengan <i>format</i> yang benar, kemudian klik Simpan	Kode Akun (400) Nama Akun (pendapatan 1) Jenis Akun (harta) Saldo Normal (debit)	Sistem akan menampilkan pesan "Format [Nama Akun] Tidak Valid"	Sesuai Harapan	<i>Valid</i>
4	Semua <i>field</i> diisi dengan benar kemudian klik Simpan	Kode Akun (400) Nama Akun (pendapatan usaha) Jenis Akun (harta) Saldo Normal (debit)	Sistem akan menerima dan menyimpan data kemudian menampilkan pesan "Data berhasil disimpan"	Sesuai Harapan	<i>Valid</i>
5	Memilih <i>field</i> kemudian klik Ubah dengan mengubah kode akun	Kode Akun (401) Nama Akun (pendapatan usaha) Jenis Akun (harta) Saldo Normal (debit)	Sistem akan menampilkan pesan "Data gagal diubah"	Sesuai Harapan	<i>Valid</i>

3. *Blackbox Testing* Form Pendapatan

Tabel III.26
***Blackbox Testing* Form Pendapatan**

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Kode Pasien tidak dipilih kemudian klik tombol Simpan	No Pendapatan (otomatis) Tanggal Pendapatan (otomatis) Kode Pasien (kosong) Daftar Pemeriksaan (dipilih) Jumlah (1)	Sistem akan menampilkan pesan "Pasien belum dipilih"	Sesuai Harapan	Valid
2	Kode Pemeriksaan tidak dipilih kemudian klik tombol Simpan	No Pendapatan (otomatis) Tanggal Pendapatan (otomatis) Kode Pasien (Dipilih) Daftar Pemeriksaan (Kosong) Jumlah (1)	Sistem akan menolak dan menampilkan pesan "Belum Ada Pemeriksaan Laboratorium Yang Dipilih"	Sesuai Harapan	Valid
3	Semua <i>field</i> diisi.	No Pendapatan (otomatis) Tanggal Pendapatan (otomatis) Kode Pasien (dipilih) Kode Pemeriksaan (dipilih) Jumlah (1)	Sistem akan menampilkan pesan "Penerimaan Pendapatan Berhasil"	Sesuai Harapan	Valid

3.5.3. Spesifikasi Hardware dan Software

Spesifikasi Hardware dan Software ini disarankan untuk dapat menggunakan sistem informasi yang telah dibuat oleh penulis sehingga bisa berjalan dengan baik .

Adapun pada tabel III.29 sebagai berikut:

Tabel III.29
Spesifikasi Hardware dan Software

Kebutuhan	Keterangan
Sistem	Windows 7 atau di atasnya
Processor	Intel Dual Core
RAM	2 GB disarankan 4GB
Harddisk	100 GB untuk instalasi sistem operasi beserta aplikasi pendukungnya
Monitor	14" LCD
Keyboard	102 Keys
Printer	Inkjet
Mouse	Standar
Software	Kebutuhan menjalankan aplikasi secara lokal Bahasa Pemrograman: Java DBMS: MySQL Report: Jasper Report dengan library IReport