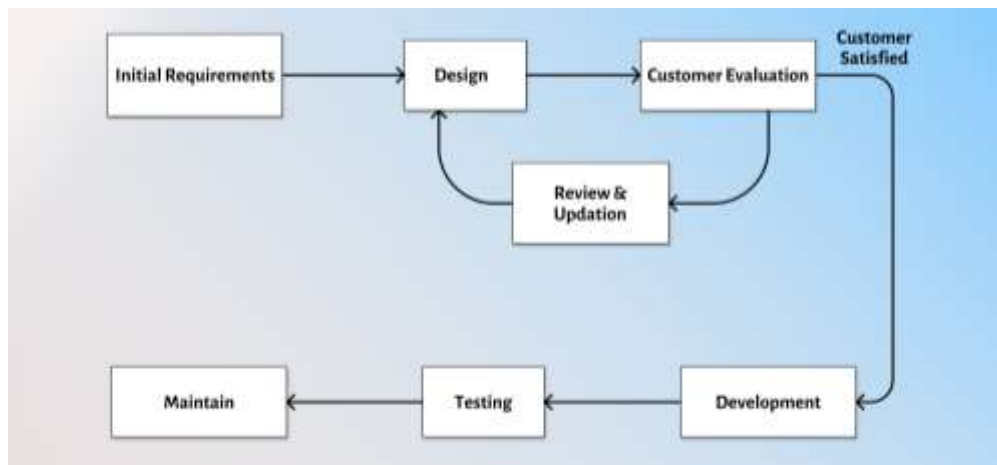


## BAB 3

### MODEL SISTEM

#### 3.1 Pengembangan Sistem

Sistem Informasi Poliklinik PDAM Kota Malang dikembangkan dengan metode SDLC prototipe. Prototipe yaitu metode pengembangan sistem yang mengizinkan pengguna memiliki gambaran awal tentang program yang akan dikembangkan serta melakukan pengujian awal. Metode ini menyajikan gambaran lengkap dari suatu sistem perangkat lunak. Prototipe terdiri dari beberapa langkah yaitu mengidentifikasi kebutuhan pengguna, merancang prototipe, evaluasi prototipe oleh pelanggan, memperbaiki dan memperbarui prototipe, pengkodean sistem, pengujian sistem, dan pemeliharaan sistem.



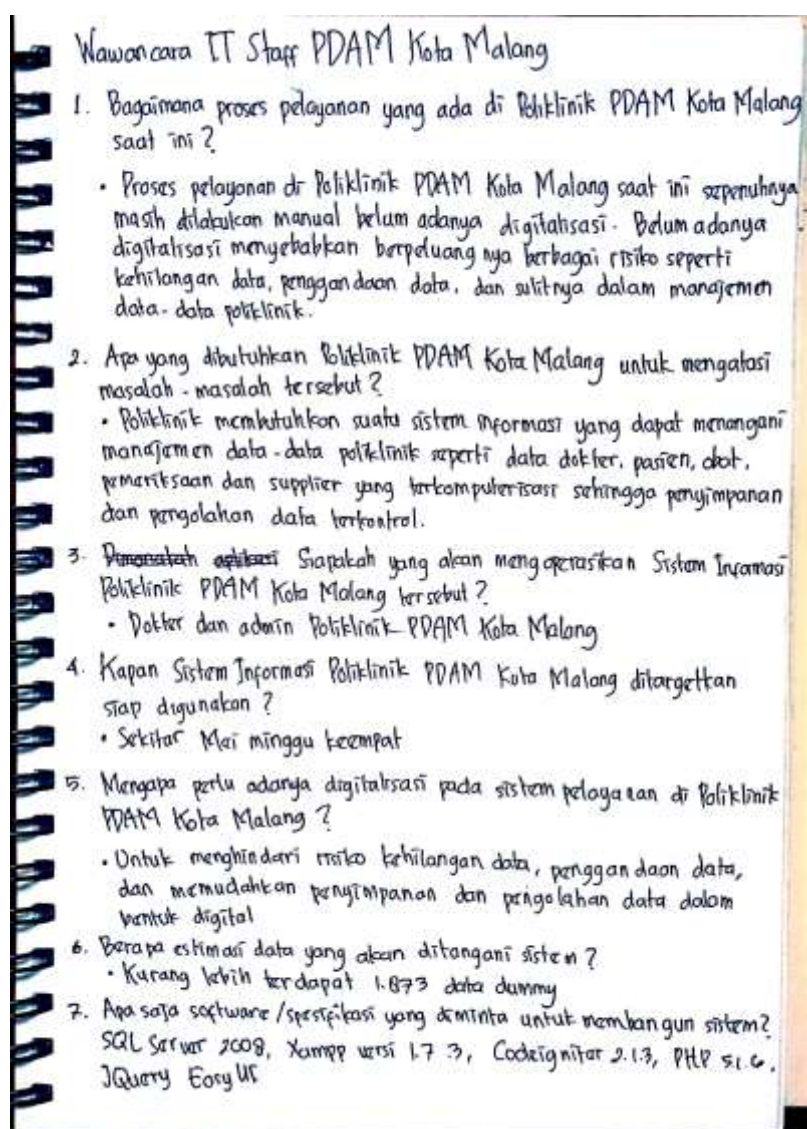
Gambar 3. 1 *SDLC Protoyping*

##### 3.1.1 Identifikasi Kebutuhan Pengguna

Identifikasi kebutuhan pengguna dilakukan melalui pertemuan secara langsung dengan pengguna sistem. Pertemuan tersebut berlokasi di PDAM Kota Malang bersama dengan IT *Staff* PDAM Kota Malang. Pengumpulan informasi mengenai rancangan sistem yang dibutuhkan pengguna dan data-data yang diperlukan dilakukan melalui wawancara dan *Focus Group Discussion* (FGD).

Wawancara adalah salah satu kaedah mengumpulkan data yang paling biasa digunakan dalam suatu penelitian. Kaedah ini digunakan ketika subjek kajian

(responden) dan peneliti berada langsung bertatap muka dalam proses mendapatkan informasi bagi keperluan data primer. Wawancara dilakukan dengan *IT Staff* PDAM Kota Malang pada tanggal 4 Januari 2021 di PDAM Kota Malang. Wawancara dilakukan untuk mengetahui kebutuhan instansi terkait sistem informasi poliklinik PDAM Kota Malang. Dari hasil wawancara tersebut dapat diketahui mengenai pengguna sistem, data-data yang dibutuhkan, spesifikasi mengenai bahasa pemrograman dan *tools* yang digunakan untuk membangun sistem.



Gambar 3. 2 Hasil Wawancara dengan *IT Staff* PDAM Kota Malang

Sedangkan metode FGD merupakan variasi dari metode wawancara yang melibatkan lebih dari satu informan. Format ideal dari FGD adalah dilakukan dalam satu meja antara para informan dan peneliti. Pelaksanaan FGD Pada praktiknya dilakukan dengan cara diskusi dimana satu orang berperan sebagai moderator. Moderator berasal dari kalangan peneliti sendiri yang mengajukan pertanyaan tanpa mengintervensi jawaban setiap informan. FGD dilakukan pada tanggal 11 Januari 2021 dengan pihak instansi yakni *IT Staff* dan humas PDAM Kota Malang. Hasil dari FGD Tersebut diperoleh penjelasan lebih detail mengenai aktor dan alur dari sistem, desain antarmuka sistem, serta data *dummy* dari data-data yang diperlukan.

### **3.1.2 Perancangan Prototipe**

Dari hasil identifikasi kebutuhan pengguna, selanjutnya dilakukan perancangan prototipe yang meliputi pembuatan bisnis proses sistem, *use case* diagram, diagram alir, rancangan basis data, dan rancangan antarmuka pengguna.

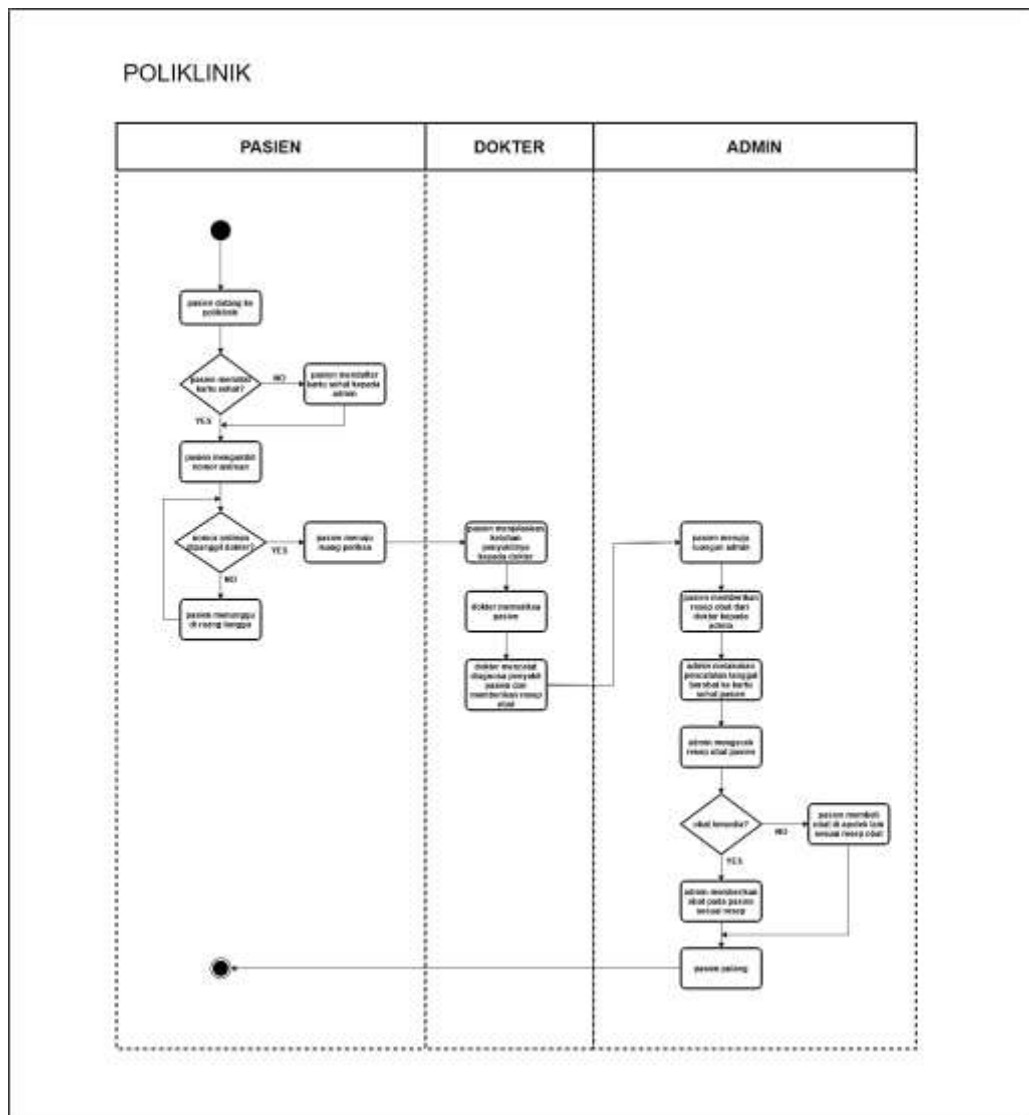
#### **3.1.2.1 Bisnis Proses**

##### **3.1.2.1.1 Bisnis Proses Saat Ini**

Gambar *Business Process Diagram* di bawah ini menunjukkan proses pengobatan poliklinik pada pasien yang termasuk karyawan maupun pensiunan PDAM Kota Malang sebelum adanya digitalisasi. Proses tersebut diawali dengan kedatangan pasien ke poliklinik, lalu pasien melihat jadwal dokter umum atau dokter gigi yang bertugas saat itu sesuai tujuannya.

Selanjutnya, pasien akan dimintai kartu sehat oleh admin poliklinik. Jika pasien tidak memiliki kartu sehat, maka pasien harus mendaftarkan kartu sehat terlebih dahulu kepada admin. Setelah itu, maka admin akan memberikan nomor antrean berobat pada pasien. Pasien dapat menunggu nomor antreannya dipanggil di ruang tunggu. Ketika dokter telah memanggil nomor antrean pasien, pasien dapat memasuki ruangan dokter sesuai dokter yang dituju yaitu dokter umum atau dokter gigi untuk mendapat penanganan lebih lanjut. Di ruang dokter, pasien dapat menyampaikan keluhan kesehatannya. Lalu dokter akan memeriksa pasien dan mendiagnosa penyakitnya. Selanjutnya, dokter akan meresepkan obat kepada pasien, yang mana resep obat tersebut dapat ditukarkan dengan obat oleh admin.

Setelah mendapat resep obat, pasien harus menyerahkan resep tersebut kepada admin. Admin akan mengecek resep obat tersebut, dan menuliskan tanggal pemeriksaan pada kartu sehat pasien. Sebelum memberikan obat kepada pasien, admin perlu mengecek ketersediaan obat terlebih dahulu di poliklinik. Jika stok obat yang diperlukan pasien habis, maka admin dapat menyarankan kepada pasien agar membeli obat di apotek luar sesuai dengan resep obat yang telah diberikan oleh dokter. Akan tetapi, jika stok obat yang diperlukan pasien masih tersedia di poliklinik, maka admin dapat memberikan obat tersebut kepada pasien beserta dengan dosis yang telah dianjurkan oleh dokter. Pembayaran biaya obat sudah termasuk dalam gaji karyawan.



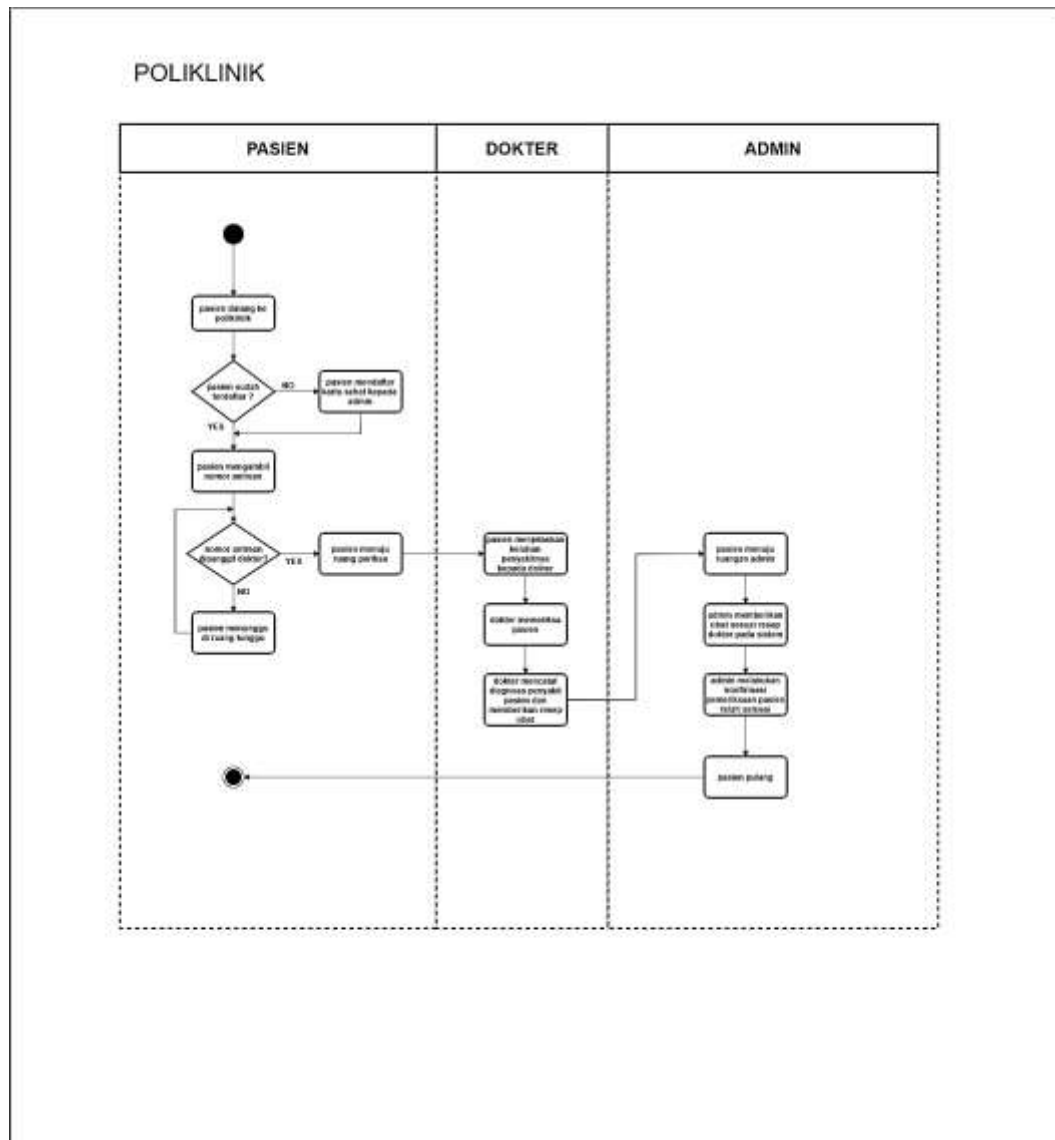
Gambar 3. 3 Diagram Bisnis Proses Saat Ini

### 3.1.2.1.2 Bisnis Proses Usulan

Setelah dilakukan proses digitalisasi pada sistem informasi Poliklinik PDAM Kota Malang, seluruh proses pendaftaran maupun pengobatan yang ada di poliklinik tersebut akan dilakukan secara digital. Gambar *Business Process Diagram* di bawah ini menunjukkan proses pengobatan pasien pada Poliklinik PDAM Kota Malang dengan media komputer. Proses tersebut tetap diawali dengan kedatangan pasien ke poliklinik dan pasien harus mengecek jadwal dokter gigi atau dokter umum yang bertugas saat itu. Akan tetapi, seluruh proses selanjutnya kecuali proses pembayaran dilakukan secara digital melalui sistem informasi yang telah dirancang.

Selanjutnya, pasien akan dimintai kartu sehat oleh admin poliklinik. Jika pasien tidak memiliki kartu sehat, maka pasien harus mendaftarkan kartu sehat terlebih dahulu kepada admin. Setelah itu, maka admin akan memberikan nomor antrean berobat pada pasien. Pasien dapat menunggu nomor antreannya dipanggil di ruang tunggu. Ketika dokter telah memanggil nomor antrean pasien, pasien dapat memasuki ruangan dokter sesuai dokter yang dituju yaitu dokter umum atau dokter gigi untuk mendapat penanganan lebih lanjut. Di ruang dokter, pasien dapat menyampaikan keluhan kesehatannya. Lalu dokter akan memeriksa pasien dan mendiagnosa penyakitnya. Selanjutnya, dokter akan meresepkan obat kepada pasien, yang mana resep obat tersebut dapat ditukarkan dengan obat oleh admin.

Setelah mendapat resep obat, pasien harus menyerahkan resep tersebut kepada admin. Admin akan mengecek resep obat tersebut, dan menuliskan tanggal pemeriksaan pada kartu sehat pasien. Kemudian, admin akan memberikan obat tersebut kepada pasien beserta dengan dosis yang telah dianjurkan oleh dokter. Pembayaran biaya obat sudah termasuk dalam gaji karyawan.

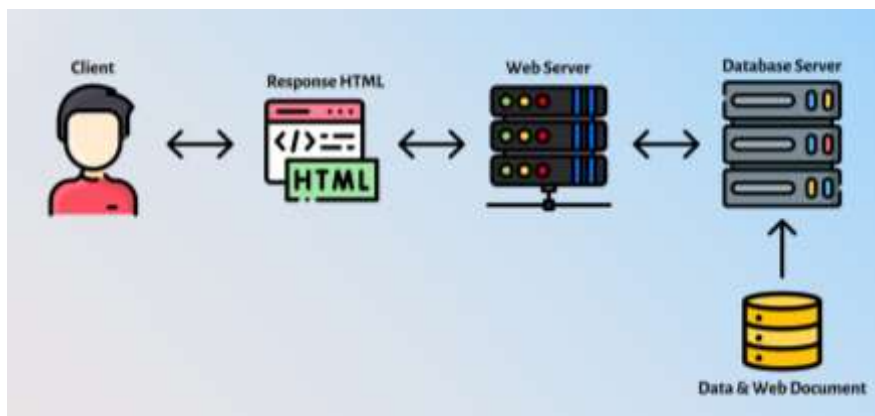


Gambar 3. 4 Diagram Bisnis Proses Usulan

### 3.1.2.2 Arsitektur dan Desain Usulan

#### 3.1.2.2.1 Arsitektur Sistem

Dalam arsitektur “**Sistem Informasi Poliklinik PDAM Kota Malang**”, ketika *client* (admin dan dokter) melakukan *request* ke *web server*, maka *data* & *web document* akan dikirimkan ke *database server*. Lalu, data yang di *request* oleh *client* akan ditampilkan kepada *client* melalui sebuah *response* html. Dengan begitu, *client* akan mendapat data yang diinginkan sesuai *request* nya.



Gambar 3. 5 Desain Arsitektur Sistem

Sistem dibangun sesuai dengan spesifikasi yang dibutuhkan *server* beserta dengan permintaan PDAM Kota Malang, dengan spesifikasi sebagai berikut:

- SQL Server Management Studio 2008 yang menangani kurang lebih 1.873 data berdasarkan data *dummy* yang telah diberikan oleh pihak PDAM Kota Malang. Pada *database* tersebut, isi data dalam tabel belum mencakup 100% data yang ada pada PDAM Kota Malang. Akan tetapi, struktur tabel sama dengan tabel pada *database* aslinya. Penggunaan SQL Server juga didasarkan pada permintaan PDAM Kota Malang yang menggunakan SQL Server sebagai basis datanya.
- Xampp versi 1.7.3 yang berfungsi menyambungkan PC ke *server*.
- CodeIgniter versi 2.1.3.
- PHP versi 5.1.6.

### 3.1.2.2.2 Use Case Diagram

Penjelasan mengenai definisi aktor, definisi *use case*, skenario *use case*, dan diagram *use case*, ditunjukkan pada materi berikut:

#### a. Definisi Aktor

Tabel 3. 1 Tabel Definisi Aktor *Use Case*

No	Aktor	Deskripsi
1	Admin Poliklinik	Admin poliklinik adalah orang yang bertugas dan memiliki hak akses untuk melakukan operasi pengelolaan data pasien, dokter, obat, <i>supplier</i> , serta mencetak laporan data yang

		diperlukan.
2	Dokter Poliklinik	Dokter poliklinik adalah orang yang bertugas dan memiliki hak akses untuk melihat data pasien, mengelola data pemeriksaan, dan mengelola data resep obat.

### b. Definisi Use Case

Tabel 3. 2 Tabel Definisi *Use Case*

No	Use Case	Deskripsi
1	<i>Login</i>	Merupakan proses untuk melakukan <i>login</i> admin dan dokter perpustakaan.
2	Mengelola Retur Obat	Mengelola retur obat merupakan proses pengelolaan data retur obat yang meliputi memasukkan retur obat, melihat retur obat, mengubah retur obat, menghapus retur obat dan mencari retur obat.
3	Mengelola Data Dokter	Mengelola data dokter merupakan proses pengelolaan data dokter yang meliputi memasukkan data dokter, melihat data dokter, mengubah data dokter, menghapus data dokter dan mencari data dokter.
4	Mengelola Data Pasien	Mengelola data pasien merupakan proses pengelolaan data pasien yang meliputi memasukkan data pasien, melihat data pasien, mengubah data pasien, menghapus data pasien dan mencari data pasien.
5	Mengelola Data <i>Supplier</i>	Mengelola data <i>supplier</i> merupakan proses pengelolaan data <i>supplier</i> yang meliputi memasukkan data <i>supplier</i> , melihat data



		<i>supplier</i> , mengubah data <i>supplier</i> , menghapus data <i>supplier</i> dan mencari data <i>supplier</i> .
<b>6</b>	Mengelola Stok Obat	Mengelola stok obat merupakan proses pengelolaan data stok obat yang meliputi memasukkan stok obat, melihat stok obat, mengubah stok obat, menghapus stok obat dan mencari stok obat.
<b>7</b>	Mengelola Surat Kontrak Dokter	Mengelola surat kontrak dokter merupakan proses pengelolaan data surat kontrak dokter yang meliputi memasukkan surat kontrak dokter, melihat surat kontrak dokter, mengubah surat kontrak dokter, menghapus surat kontrak dokter dan mencari surat kontrak dokter.
<b>8</b>	Melihat Data Penyakit	Merupakan proses menampilkan data penyakit yang ada di dalam basis data.
<b>9</b>	Membuat Laporan Obat Masuk	Merupakan proses memasukkan data obat masuk ke dalam laporan ketika ada obat masuk dari <i>supplier</i> .
<b>10</b>	Membuat Laporan Obat Rusak/Kadaluarsa	Merupakan proses memasukkan data obat rusak ke dalam laporan ketika ada obat yang rusak.
<b>11</b>	Membuat Laporan Periksa	Merupakan proses memasukkan data periksa ke dalam laporan.
<b>12</b>	Membuat Laporan Obat Keluar	Merupakan proses memasukkan data obat keluar ke dalam laporan ketika ada obat yang keluar.
<b>13</b>	Mencetak Laporan	Merupakan proses mencetak laporan dari data yang dibutuhkan.

14	Memasukkan Data Pemeriksaan	Merupakan proses dokter memasukkan data hasil pemeriksaan pasien kedalam basis data.
15	Melihat Data Pasien	Merupakan proses dokter menampilkan data pasien yang ada di dalam basis data.
16	Melihat Riwayat Penyakit	Merupakan proses dokter menampilkan data riwayat penyakit yang ada di dalam basis data.
17	Mengelola Resep Obat	Mengelola resep obat oleh dokter merupakan proses pengelolaan data resep obat yang meliputi memasukkan resep obat, melihat resep obat, mengubah resep obat, menghapus resep obat dan mencari resep obat.

**c. Skenario Use Case**

Nama *Use Case*: *Login*

Aktor: Admin dan Dokter

Skenario:

Tabel 3. 3 Tabel Skenario *Use Case Login*

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
Memasukkan <i>username</i> dan <i>password</i> .	
	Memeriksa valid tidaknya data masukan dengan memeriksa ke tabel karyawan.
	Masuk ke aplikasi sistem informasi poliklinik.
<b>Skenario Alternatif</b>	
Memasukkan <i>username</i> dan <i>password</i> .	
	Memeriksa valid tidaknya data masukan dengan memeriksa ke tabel karyawan.
	Menampilkan pesan <i>login</i> tidak valid.

Memasukkan <i>username</i> dan <i>password</i> yang valid.	
	Memeriksa valid tidaknya data masukan dengan memeriksa ke tabel karyawan.
	Masuk ke aplikasi sistem informasi poliklinik.

Nama *Use Case*: Mengelola Retur Obat

Aktor: Admin

Skenario:

Tabel 3. 4 Tabel Skenario Mengelola Retur Obat

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .
Mengelola data retur obat.	
	Memeriksa valid tidaknya data yang diolah.
	Menyimpan perubahan data yang diolah ke basis data.
	Menampilkan pesan pengelolaan data berhasil.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .
Mengelola data retur obat.	
	Memeriksa valid tidaknya data yang diolah.
	Mengeluarkan pesan data yang dikelola tidak valid
Memperbaiki data retur obat yang tidak valid.	
	Memeriksa valid tidaknya data yang

	diolah.
	Menyimpan perubahan data yang diolah ke basis data.
	Menampilkan pesan pengelolaan data berhasil.

Nama *Use Case*: Mengelola Data Dokter

Aktor: Admin

Skenario:

Tabel 3. 5 Tabel Skenario Mengelola Data Dokter

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .
Mengelola data dokter.	
	Memeriksa valid tidaknya data yang diolah.
	Menyimpan perubahan data yang diolah ke basis data.
	Menampilkan pesan pengelolaan data berhasil.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .
Mengelola data dokter.	
	Memeriksa valid tidaknya data yang diolah.
	Mengeluarkan pesan data yang dikelola tidak valid
Memperbaiki data dokter yang tidak valid.	
	Memeriksa valid tidaknya data yang diolah.

	Menyimpan perubahan data yang diolah ke basis data.
	Menampilkan pesan pengelolaan data berhasil.

Nama *Use Case*: Mengelola Data Pasien

Aktor: Admin

Skenario:

Tabel 3. 6 Tabel Skenario Mengelola Data Pasien

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .
Mengelola data pasien.	
	Memeriksa valid tidaknya data yang diolah.
	Menyimpan perubahan data yang diolah ke basis data.
	Menampilkan pesan pengelolaan data berhasil.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .
Mengelola data pasien.	
	Memeriksa valid tidaknya data yang diolah.
	Mengeluarkan pesan data yang dikelola tidak valid
Memperbaiki data pasien yang tidak valid.	
	Memeriksa valid tidaknya data yang diolah.
	Menyimpan perubahan data yang diolah ke basis data.

	Menampilkan pesan pengelolaan data berhasil.
--	--

Nama *Use Case*: Mengelola Data *Supplier*

Aktor: Admin

Skenario:

Tabel 3. 7 Tabel Skenario Mengelola Data *Supplier*

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .
Mengelola data <i>supplier</i> .	
	Memeriksa valid tidaknya data yang diolah.
	Menyimpan perubahan data yang diolah ke basis data.
	Menampilkan pesan pengelolaan data berhasil.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .
Mengelola data <i>supplier</i> .	
	Memeriksa valid tidaknya data yang diolah.
	Mengeluarkan pesan data yang dikelola tidak valid
Memperbaiki data <i>supplier</i> yang tidak valid.	
	Memeriksa valid tidaknya data yang diolah.
	Menyimpan perubahan data yang diolah ke basis data.
	Menampilkan pesan pengelolaan data berhasil.

Nama *Use Case*: Mengelola Data Stok Obat

Aktor: Admin

Skenario:

Tabel 3. 8 Tabel Skenario Mengelola Data Stok Obat

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .
Mengelola data stok obat.	
	Memeriksa valid tidaknya data yang diolah.
	Menyimpan perubahan data yang diolah ke basis data.
	Menampilkan pesan pengelolaan data berhasil.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .
Mengelola data stok obat.	
	Memeriksa valid tidaknya data yang diolah.
	Mengeluarkan pesan data yang dikelola tidak valid
Memperbaiki data stok obat yang tidak valid.	
	Memeriksa valid tidaknya data yang diolah.
	Menyimpan perubahan data yang diolah ke basis data.
	Menampilkan pesan pengelolaan data berhasil.

Nama *Use Case*: Mengelola Data Surat Kontrak Dokter

Aktor: Admin

Skenario:

Tabel 3. 9 Tabel Skenario Mengelola Data Surat Kontrak Dokter

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .
Mengelola data surat kontrak dokter.	
	Memeriksa valid tidaknya data yang diolah.
	Menyimpan perubahan data yang diolah ke basis data.
	Menampilkan pesan pengelolaan data berhasil.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .
Mengelola data surat kontrak dokter.	
	Memeriksa valid tidaknya data yang diolah.
	Mengeluarkan pesan data yang dikelola tidak valid
Memperbaiki data surat kontrak dokter yang tidak valid.	
	Memeriksa valid tidaknya data yang diolah.
	Menyimpan perubahan data yang diolah ke basis data.
	Menampilkan pesan pengelolaan data berhasil.

Nama *Use Case*: Melihat Data Penyakit

Aktor: Admin

Skenario:



Tabel 3. 10 Tabel Skenario Melihat Data Penyakit

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .
Memilih menu penyakit.	
	Menunggu proses menampilkan menu penyakit.
	Sistem menampilkan data penyakit yang dibuka.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .
Memilih menu penyakit.	
	Menunggu proses menampilkan menu penyakit.
	Sistem gagal menampilkan data penyakit yang dibuka.

Nama *Use Case*: Membuat Laporan Obat Masuk

Aktor: Admin

Skenario:

Tabel 3. 11 Tabel Skenario Membuat Laporan Obat Masuk

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .
Memasukkan data obat masuk sesuai kolom yang ada.	
	Memeriksa valid tidaknya data masukkan.
	Menyimpan data yang dimasukkan ke basis data.
	Menampilkan pesan penyimpanan

	data berhasil.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .
Memasukkan data obat masuk sesuai kolom yang ada.	
	Memeriksa valid tidaknya data yang dimasukkan.
	Mengeluarkan pesan data yang dimasukkan tidak valid
Memperbaiki data masukkan yang tidak valid.	
	Memeriksa valid tidaknya data yang dimasukkan.
	Menyimpan perubahan data yang dimasukkan ke basis data.
	Menampilkan pesan penyimpanan data berhasil.

Nama *Use Case*: Membuat Laporan Obat Rusak/Kadaluarsa

Aktor: Admin

Skenario:

Tabel 3. 12 Tabel Skenario Membuat Laporan Obat Rusak/Kadaluarsa

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .
Memasukkan data obat rusak sesuai kolom yang ada.	
	Memeriksa valid tidaknya data masukkan.
	Menyimpan data yang dimasukkan ke basis data.
	Menampilkan pesan penyimpanan

	data berhasil.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .
Memasukkan data obat rusak sesuai kolom yang ada.	
	Memeriksa valid tidaknya data yang dimasukkan.
	Mengeluarkan pesan data yang dimasukkan tidak valid
Memperbaiki data masukkan yang tidak valid.	
	Memeriksa valid tidaknya data yang dimasukkan.
	Menyimpan perubahan data yang dimasukkan ke basis data.
	Menampilkan pesan penyimpanan data berhasil.

Nama *Use Case*: Membuat Laporan Periksa

Aktor: Admin

Skenario:

Tabel 3. 13 Tabel Skenario Membuat Laporan Periksa

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .
Memasukkan data laporan periksa sesuai kolom yang ada.	
	Memeriksa valid tidaknya data masukkan.
	Menyimpan data yang dimasukkan ke basis data.
	Menampilkan pesan penyimpanan

	data berhasil.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .
Memasukkan data laporan periksa sesuai kolom yang ada.	
	Memeriksa valid tidaknya data yang dimasukkan.
	Mengeluarkan pesan data yang dimasukkan tidak valid
Memperbaiki data masukkan yang tidak valid.	
	Memeriksa valid tidaknya data yang dimasukkan.
	Menyimpan perubahan data yang dimasukkan ke basis data.
	Menampilkan pesan penyimpanan data berhasil.

Nama *Use Case*: Mencetak Laporan

Aktor: Admin

Skenario:

Tabel 3. 14 Tabel Skenario Mencetak Laporan

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .
Memilih menu cetak laporan.	
	Menunggu proses cetak laporan.
	Sistem mencetak data laporan yang dipilih.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .

Memilih menu cetak laporan.	
	Menunggu proses cetak laporan.
	Sistem gagal mencetak data laporan yang dipilih.

Nama *Use Case*: Memasukkan Data Pemeriksaan

Aktor: Dokter

Skenario:

Tabel 3. 15 Tabel Skenario Memasukkan Data Pemeriksaan

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .
Memasukkan data pemeriksaan sesuai kolom yang ada.	
	Memeriksa valid tidaknya data masukkan.
	Menyimpan data yang dimasukkan ke basis data.
	Menampilkan pesan penyimpanan data berhasil.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .
Memasukkan data pemeriksaan sesuai kolom yang ada.	
	Memeriksa valid tidaknya data yang dimasukkan.
	Mengeluarkan pesan data yang dimasukkan tidak valid
Memperbaiki data masukkan yang tidak valid.	
	Memeriksa valid tidaknya data yang dimasukkan.

	Menyimpan perubahan data yang dimasukkan ke basis data.
	Menampilkan pesan penyimpanan data berhasil.

Nama *Use Case*: Melihat Data Pasien

Aktor: Dokter

Skenario:

Tabel 3. 16 Tabel Skenario Melihat Data Pasien

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .
Memilih menu pasien.	
	Menunggu proses menampilkan menu pasien.
	Sistem menampilkan data pasien yang dibuka.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .
Memilih menu pasien.	
	Menunggu proses menampilkan menu pasien.
	Sistem gagal menampilkan data pasien yang dibuka.

Nama *Use Case*: Melihat Data Riwayat Penyakit

Aktor: Dokter

Skenario:

Tabel 3. 17 Tabel Skenario Melihat Data Riwayat Penyakit

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .

Memilih menu penyakit.	
	Menunggu proses menampilkan menu penyakit.
	Sistem menampilkan data penyakit yang dibuka.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .
Memilih menu penyakit.	
	Menunggu proses menampilkan menu penyakit.
	Sistem gagal menampilkan data penyakit yang dibuka.

Nama *Use Case*: Mengelola Data Pemeriksaan

Aktor: Dokter

Skenario:

Tabel 3. 18 Tabel Skenario Mengelola Data Pemeriksaan

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	Memeriksa status <i>login</i> .
Mengelola data pemeriksaan.	
	Memeriksa valid tidaknya data yang diolah.
	Menyimpan perubahan data yang diolah ke basis data.
	Menampilkan pesan pengelolaan data berhasil.
<b>Skenario Alternatif</b>	
	Memeriksa status <i>login</i> .
Mengelola data pemeriksaan.	
	Memeriksa valid tidaknya data yang

	diolah.
	Mengeluarkan pesan data yang dikelola tidak valid
Memperbaiki data pemeriksaan yang tidak valid.	
	Memeriksa valid tidaknya data yang diolah.
	Menyimpan perubahan data yang diolah ke basis data.
	Menampilkan pesan pengelolaan data berhasil.

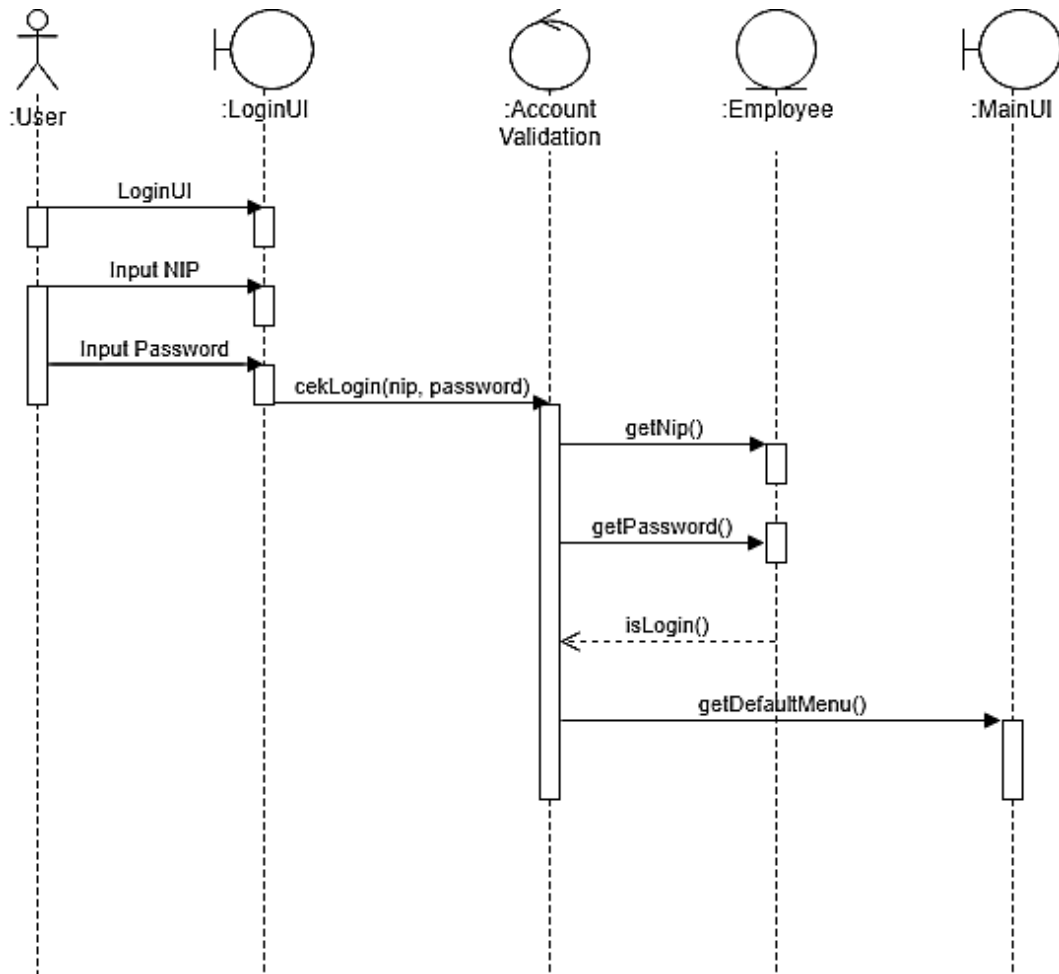
#### d. Diagram Use Case

Sistem informasi poliklinik memiliki 2 aktor yaitu admin dan dokter. Untuk mengakses sistem informasi, para aktor tersebut diharuskan untuk melakukan *login* terlebih dahulu. Setelah melakukan *login*, admin dapat melakukan beberapa tindakan seperti mengelola CRUD master data yang meliputi pengelolaan retur obat, pengelolaan data dokter, pengelolaan data pasien, pengelolaan data *supplier*, pengelolaan stok obat, pengelolaan surat kontrak dokter, dan melihat data penyakit. Admin juga dapat membuat laporan seperti pencetakan laporan, pembuatan laporan obat keluar, pembuatan laporan periksa, pembuatan laporan obat rusak atau kedaluwarsa, dan pembuatan laporan obat masuk.

Sedangkan pada sisi dokter, setelah melakukan *login* dokter dapat melakukan beberapa aksi diantaranya memasukkan data pemeriksaan, melihat data pasien beserta dengan riwayat penyakitnya, dan mengelola resep obat. *Use Case Diagram* admin dan dokter ditunjukkan pada Gambar 3.6 di bawah ini.

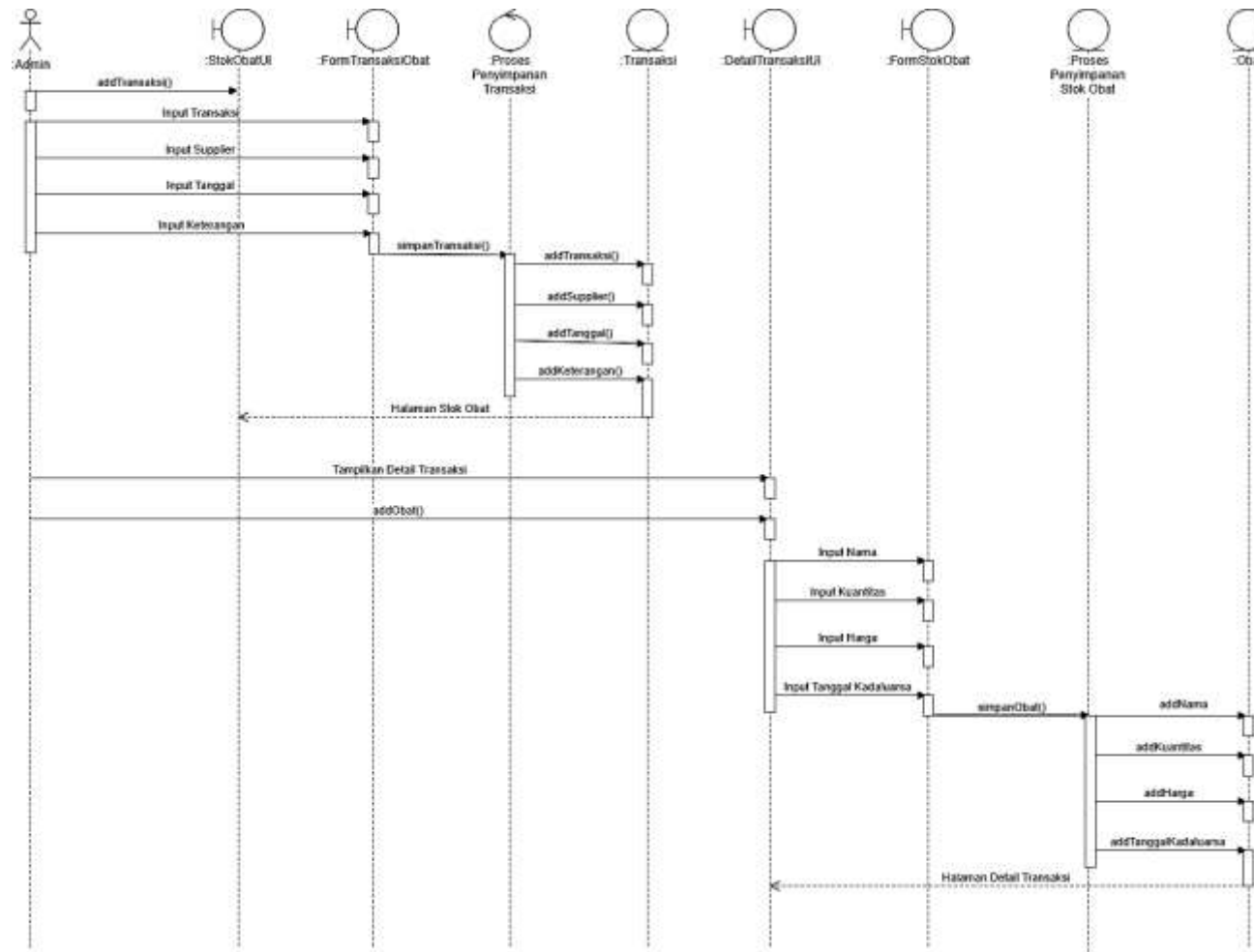






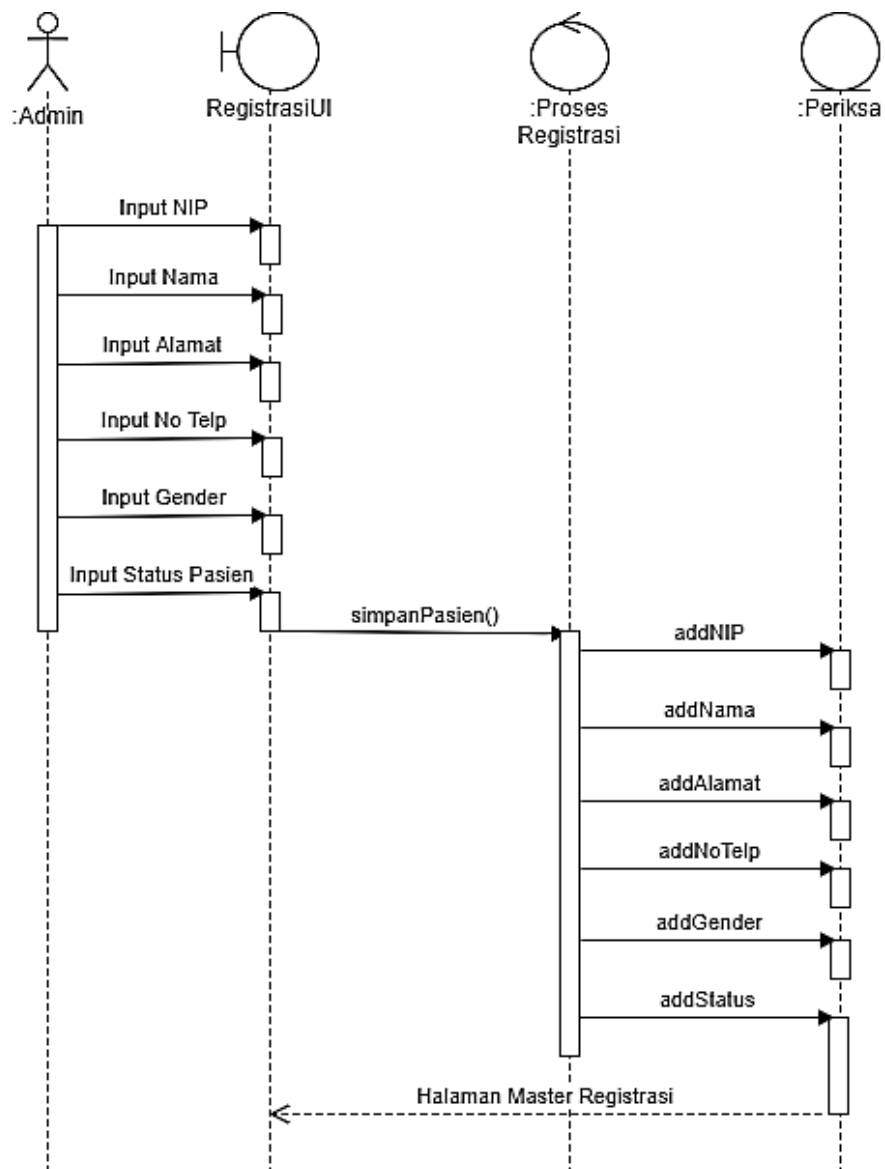
Gambar 3. 7 Sequence Diagram Login Pengguna

Setelah melakukan *login*, admin dapat melakukan proses pengolahan data stok obat untuk membeli maupun mengisi stok awal obat. Proses pengolahan obat dimulai setelah admin menginputkan data transaksi terlebih dahulu, dengan memasukkan data jenis transaksi, *supplier*, tanggal, dan keterangan ke basis data transaksi. Kemudian data tersebut akan tampil pada tabel transaksi obat, dan admin harus mengklik data tersebut untuk menampilkan detailnya. Pada halaman data detail transaksi tersebut, admin dapat menginputkan data stok obat, dengan memasukkan nama, kuantitas, harga, dan kedaluwarsa. Kemudian sistem akan memproses untuk menyimpan data tersebut ke basis data obat. Obat dengan jenis yang sama, jumlahnya akan otomatis bertambah.



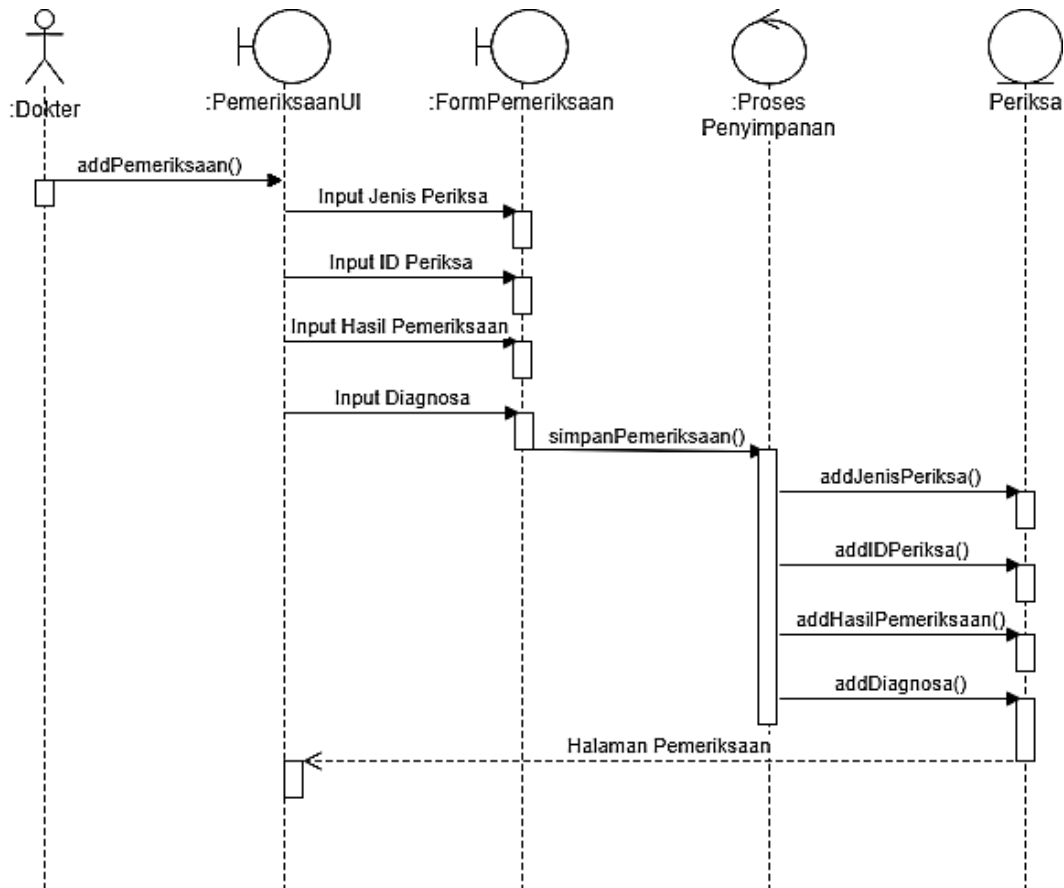
Gambar 3. 8 *Sequence Diagram* Mengelola Stok Obat

Selain manajemen data stok obat, admin juga dapat melakukan pendaftaran pasien poliklinik. Proses ini diawali dengan admin mengecek apakah pasien tersebut sudah pernah terdaftar atau belum, apabila belum terdaftar maka admin harus menambah data pasien di data master pasien dahulu. Namun, apabila pasien sudah pernah terdaftar, maka admin dapat langsung membuka *form registrasi*, kemudian admin menginputkan NIP, nama, alamat, no telp, gender, dan status pasien pada *form* tersebut. Selanjutnya, sistem akan memproses penyimpanan data ke basis data poliklinik, dan memunculkan data baru tersebut di halaman *registrasi*.



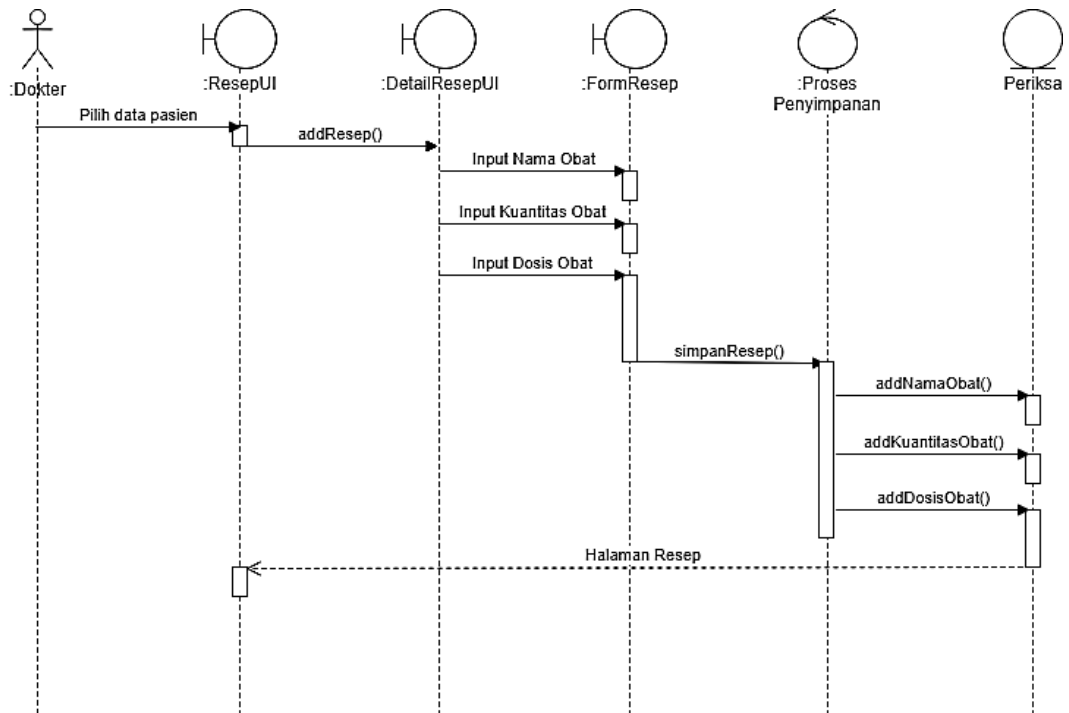
Gambar 3. 9 *Sequence Diagram* Pendaftaran Pasien

Setelah pasien mendaftar, maka pasien akan memasuki ruang pemeriksaan untuk diperiksa oleh dokter. Kemudian, dokter akan menginputkan hasil pemeriksaan ke sistem berupa jenis periksa, id periksa pasien, hasil pemeriksaan, dan diagnosa. Lalu, sistem akan menyimpan data-data tersebut ke dalam basis data, dan menampilkannya pada halaman pemeriksaan dokter dan admin.



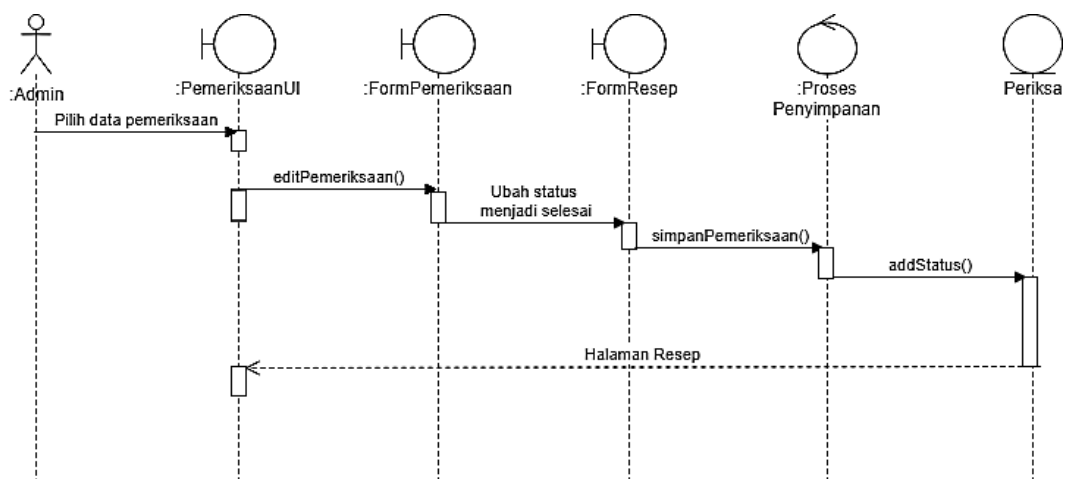
Gambar 3. 10 *Sequence Diagram* Pemeriksaan Pasien

Selain hasil pemeriksaan, dokter juga dapat menginputkan resep obat untuk pasien. Data-data yang diinputkan dokter pada sistem berupa nama obat, kuantitas obat, dan harga obat. Kemudian sistem akan menyimpan data tersebut ke basis data dan menampilkannya pada halaman resep dokter dan admin, yang selanjutnya admin akan melihat daftar resep obat tersebut dan memberikan obat yang sesuai kepada pasien



Gambar 3. 11 *Sequence Diagram* Resep Obat

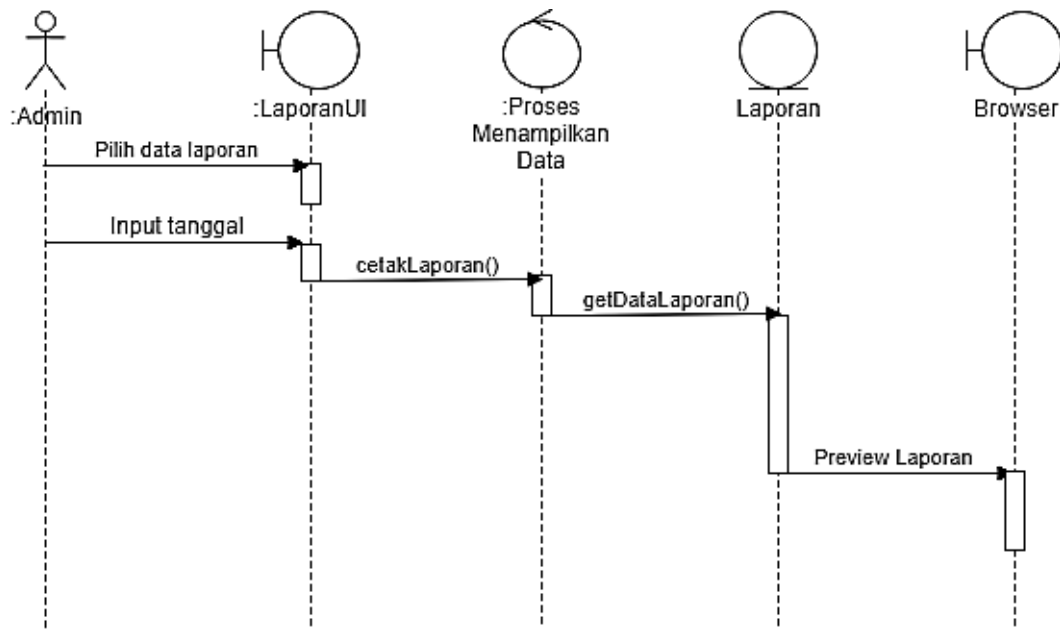
Setelah admin memberikan obat sesuai dengan resep obat dari dokter, maka admin harus mengedit status pemeriksaan pada sistem menjadi selesai, yang menandakan bahwa proses pemeriksaan pasien telah selesai.



Gambar 3. 12 *Sequence Diagram* Edit Status Pemeriksaan

Selain pengolahan data, admin juga dapat melakukan pencetakan laporan yang ada pada sistem. Terdapat beberapa data yang dapat dicetak seperti data

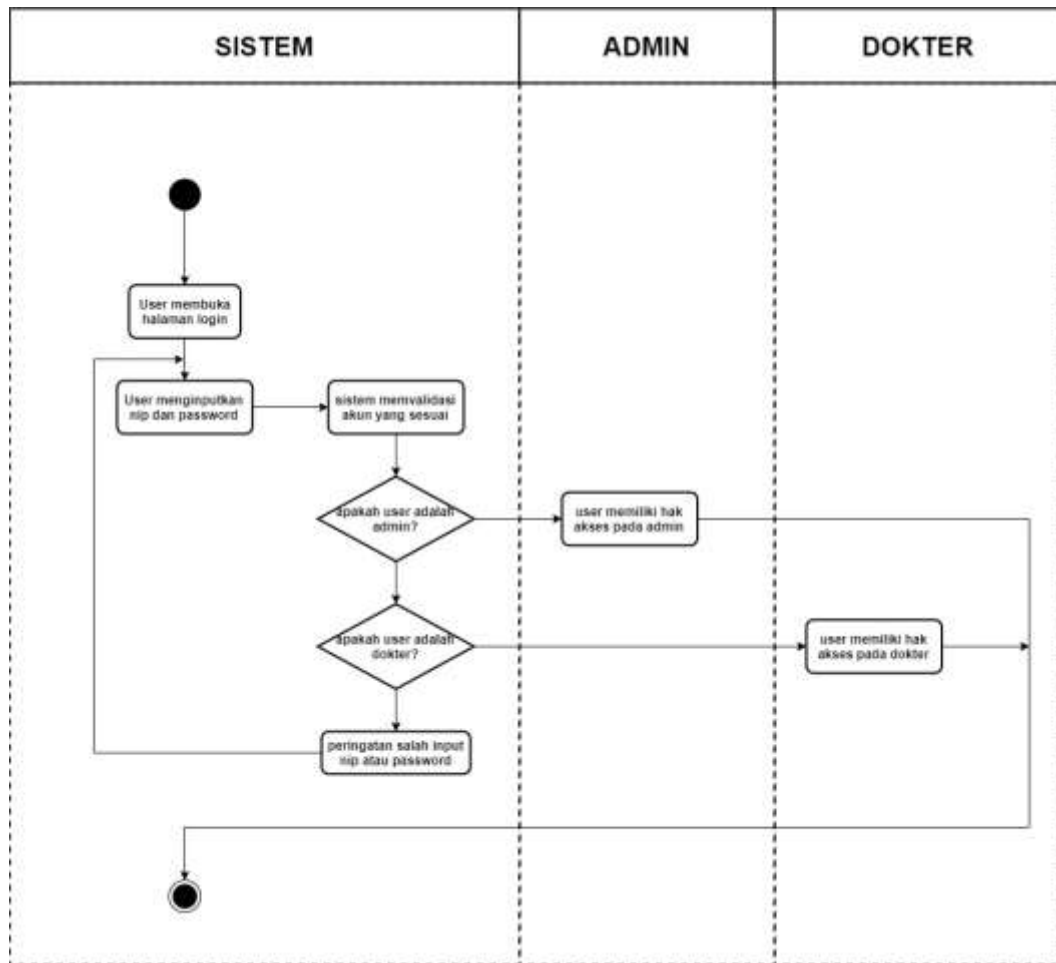
obat masuk, resep, bulanan dan data pemeriksaan. Admin dapat memilih salah satu dari data tersebut, kemudian menginputkan rentang tanggal data yang akan dicetak, kemudian mengklik tombol cetak PDF. Selanjutnya sistem akan menampilkan *preview* data yang akan dicetak admin di *browser*.



Gambar 3. 13 *Sequence Diagram* Cetak Laporan

#### 3.1.2.2.4 *Activity Diagram*

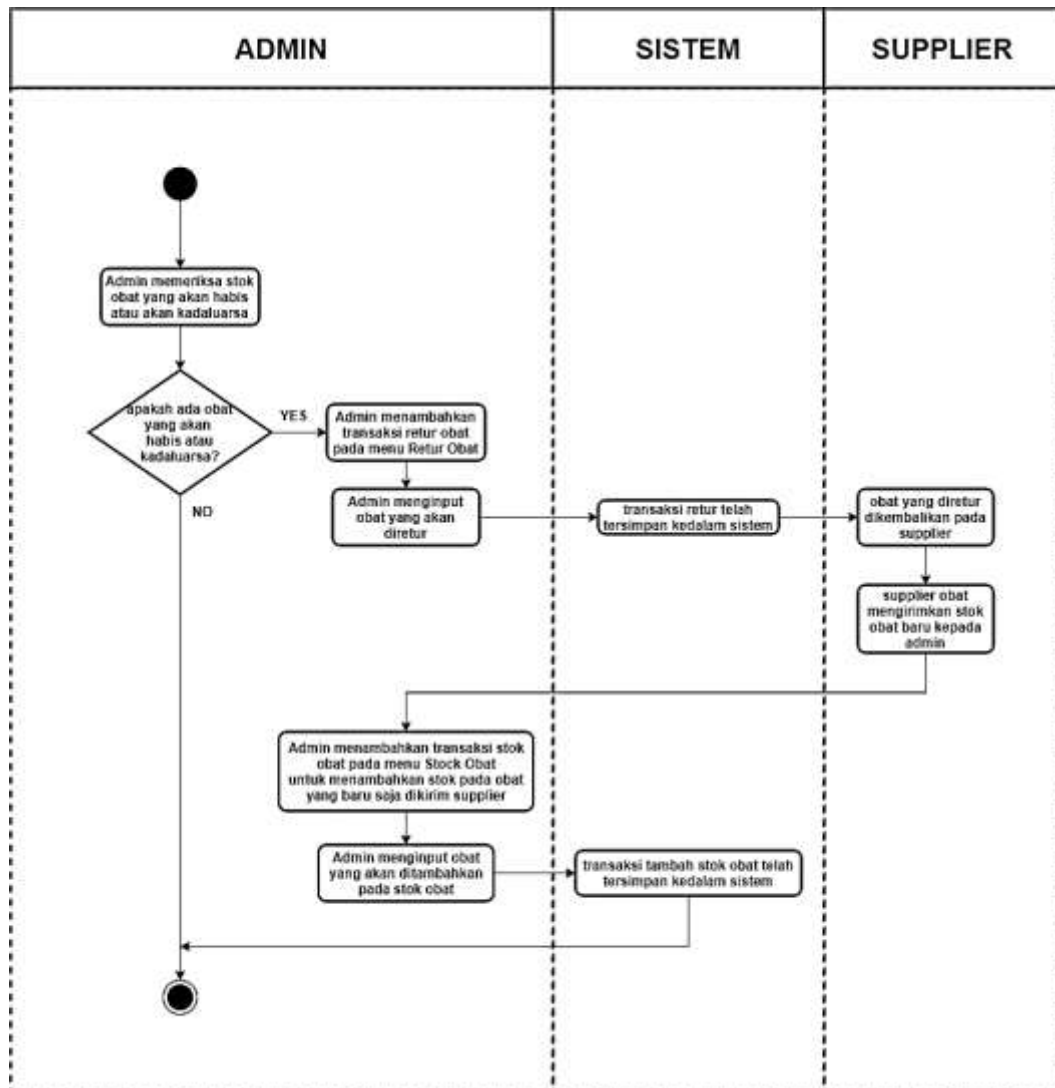
Diagram di bawah ini, menunjukkan proses *login* pengguna yaitu admin dan dokter. Pengguna memasukkan NIP dan *password* yang terdaftar pada sistem. Kemudian sistem akan memvalidasi data inputan pengguna dan mengecek apakah pengguna tersebut admin atau dokter. Selanjutnya, pengguna akan menuju ke halaman utamanya masing-masing.



Gambar 3. 14 *Activity Diagram* Login Pengguna

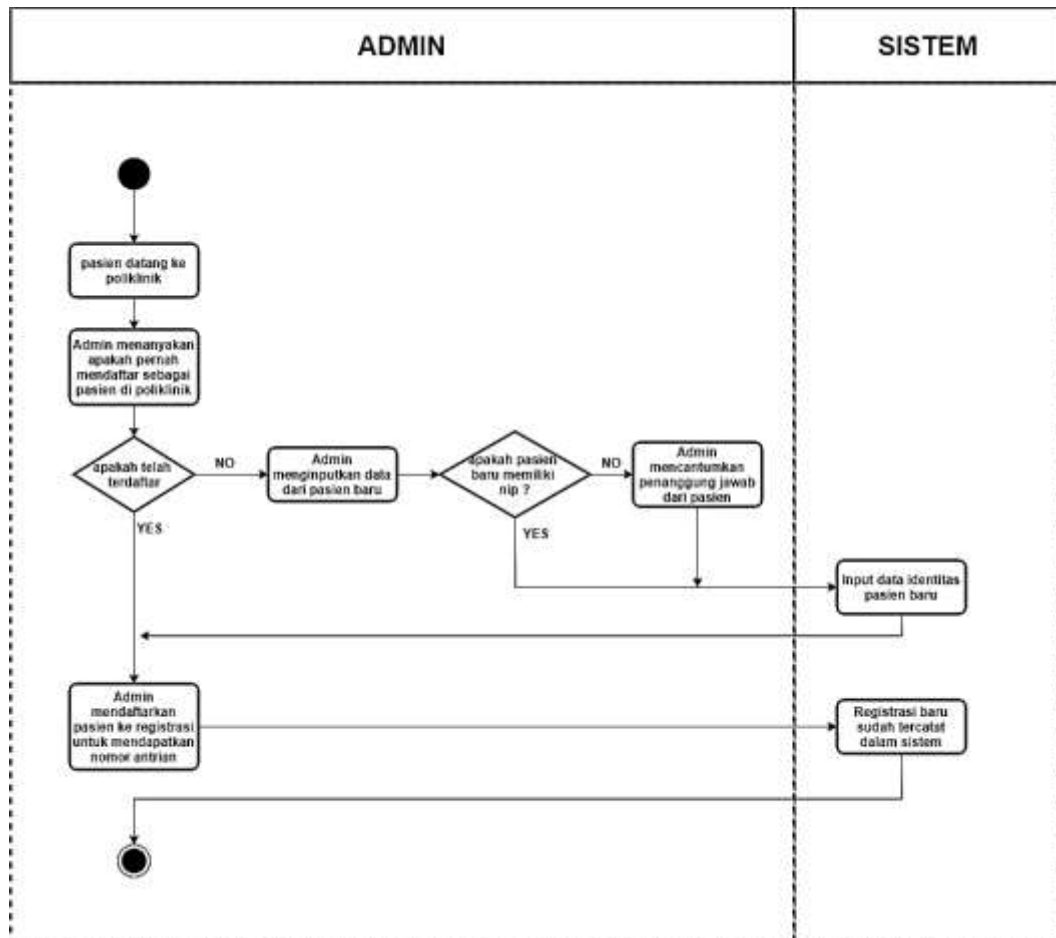
Admin dapat melakukan pengolahan obat dengan mengecek di sistem apakah ada obat yang kedaluwarsa atau tidak, jika ada obat yang kedaluwarsa, maka admin akan meretur obat tersebut kepada *supplier*. Lalu admin juga dapat mengisi kembali obat tersebut dengan membeli obat kepada *supplier* dan memasukkan data-datanya pada sistem.





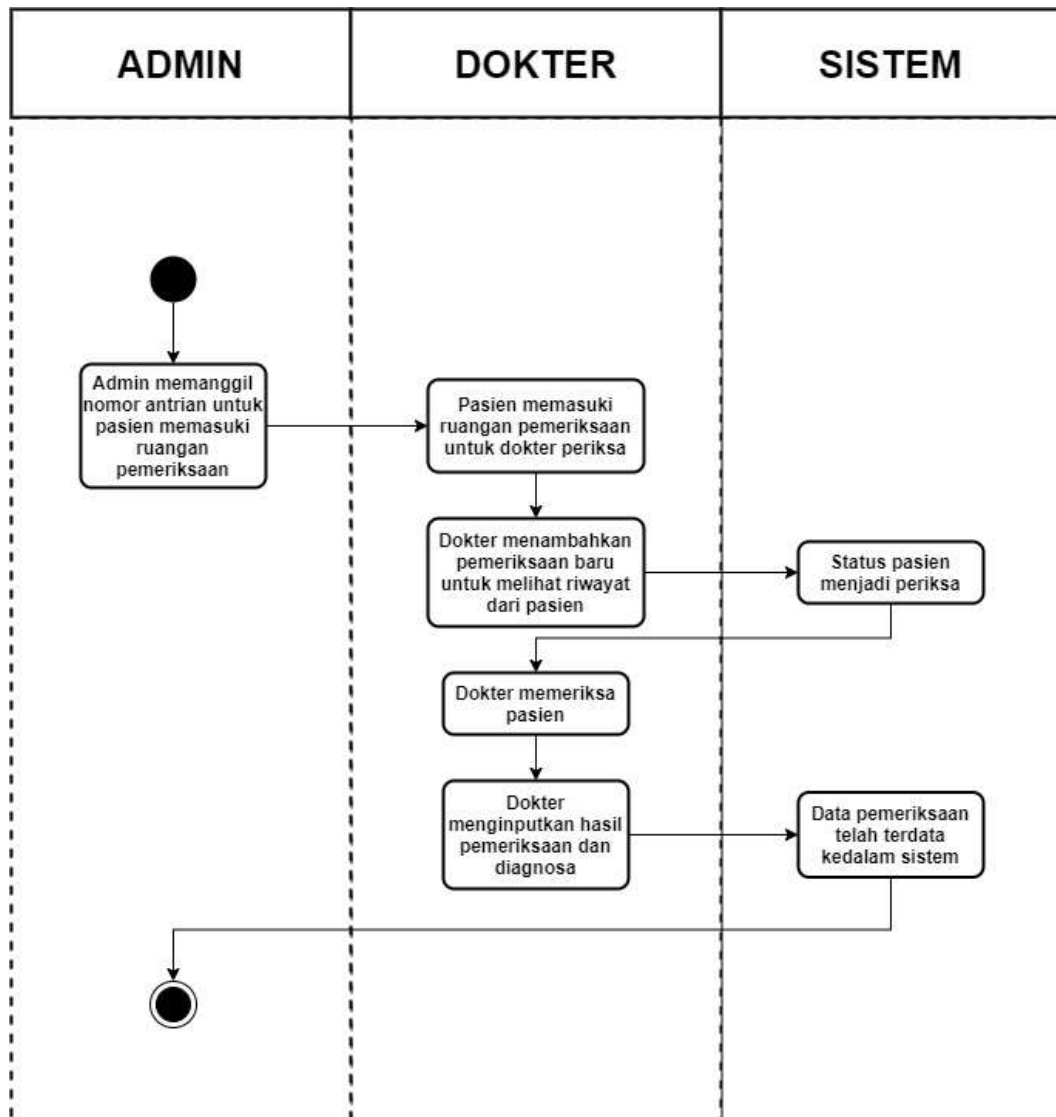
Gambar 3. 15 Activity Diagram Mengelola Stok Obat

Pada proses pemeriksaan, pertama-tama pasien harus mendaftarkan diri pada admin. Lalu admin akan mengecek apakah pasien tersebut pernah terdaftar di sistem atau belum. Jika belum, maka admin akan menginputkan data pasien tersebut terlebih dahulu dan jika pasien tidak memiliki NIP yang artinya pasien tersebut adalah non pegawai atau pensiunan, maka NIP dapat dikosongi dan dicantumkan data penanggungjawab. Setelah pasien terdaftar, maka pasien akan mendapat nomor antrian pemeriksaan.



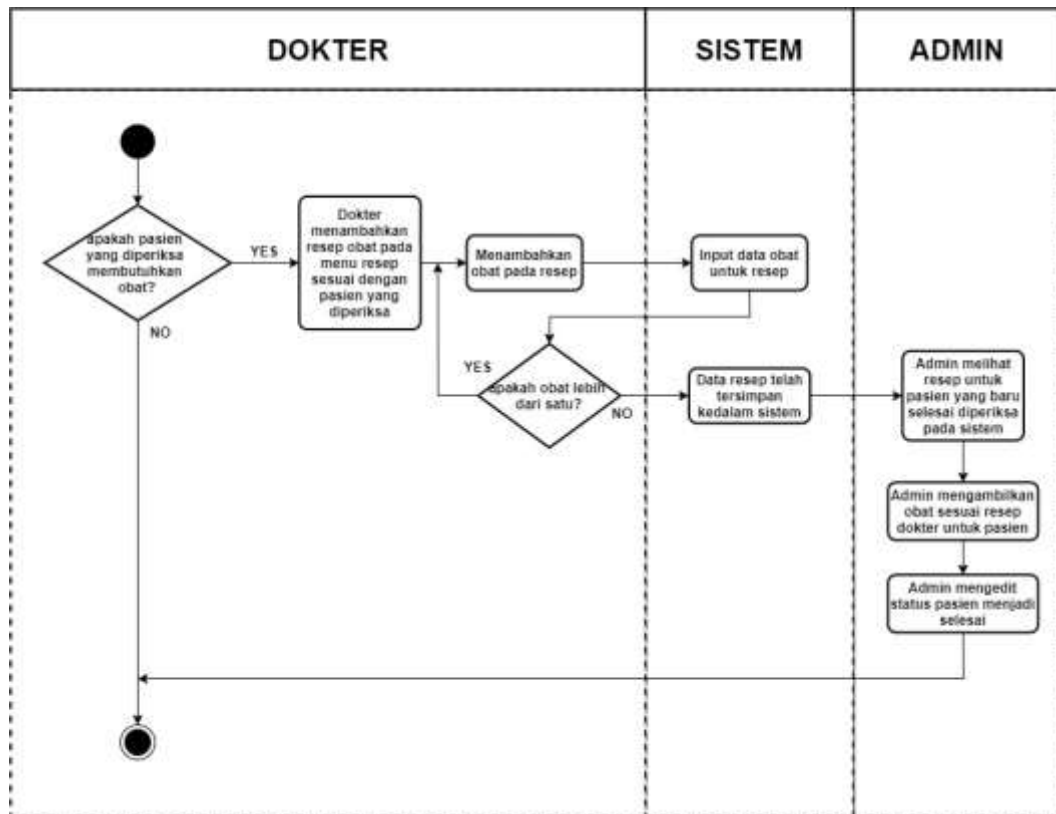
Gambar 3. 16 Activity Diagram Pendaftaran Pasien

Setelah mendapat nomor antrian, maka pasien harus menunggu sampai nomor antrian tersebut dipanggil oleh admin. Kemudian pasien dapat memasuki ruang pemeriksaan untuk diperiksa dokter. Lalu dokter menginputkan hasil pemeriksaan dan diagnosa pasien pada sistem. Pada proses ini, status pemeriksaan pasien masih tetap periksa.



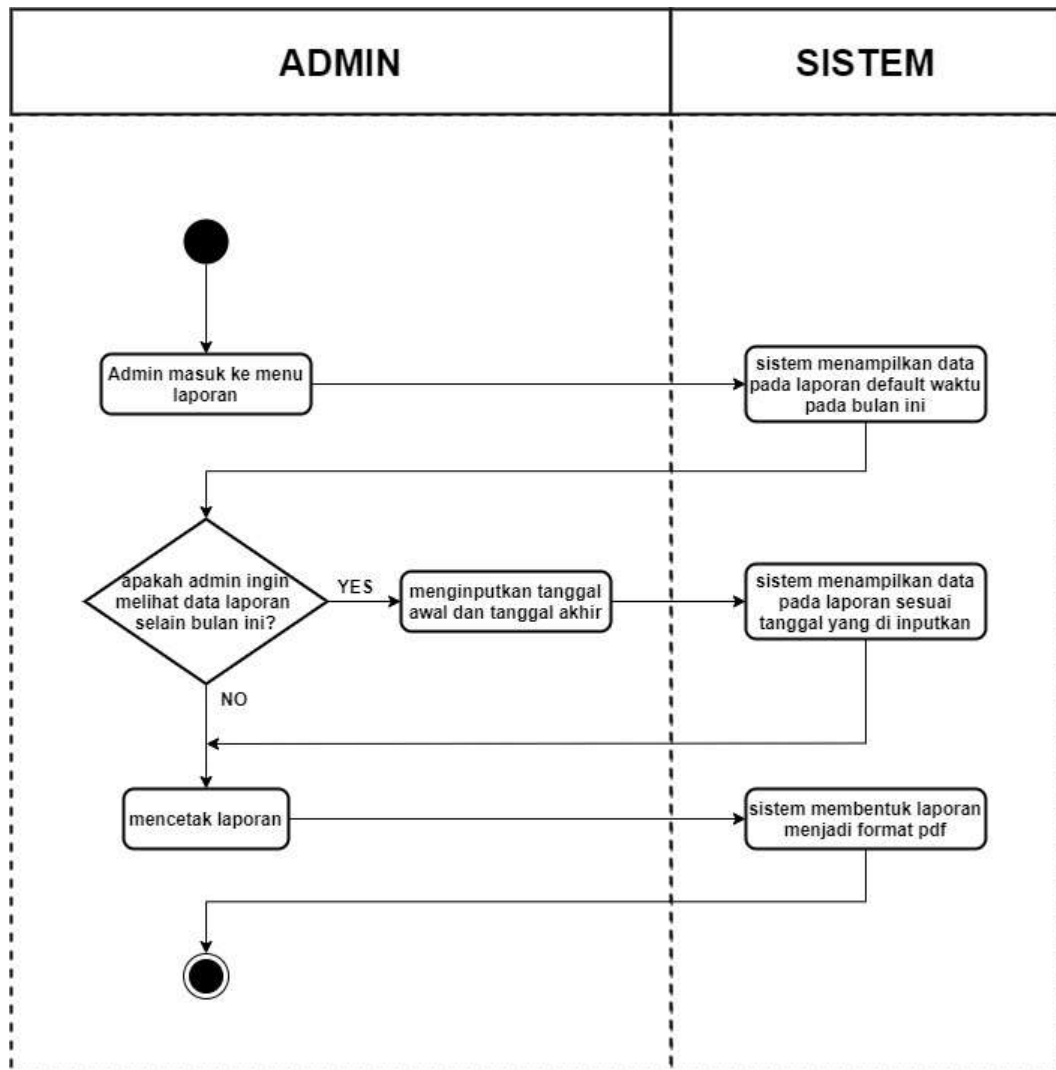
Gambar 3. 17 *Activity Diagram* Pemeriksaan Pasien

Sehabis melakukan pemeriksaan, dokter juga dapat menginputkan data resep obat untuk pasien jika dibutuhkan. Selanjutnya, data resep tersebut akan tersimpan ke dalam basis data dan ditampilkan pada halaman resep dokter dan admin. Kemudian admin akan melihat resep obat tersebut, dan memberikan obat yang sesuai kepada pasien. Lalu, admin akan mengedit status pemeriksaan pasien menjadi selesai, yang menandakan bahwa proses pemeriksaan pasien di poliklinik telah selesai.



Gambar 3. 18 Activity Diagram Mengelola Resep Obat

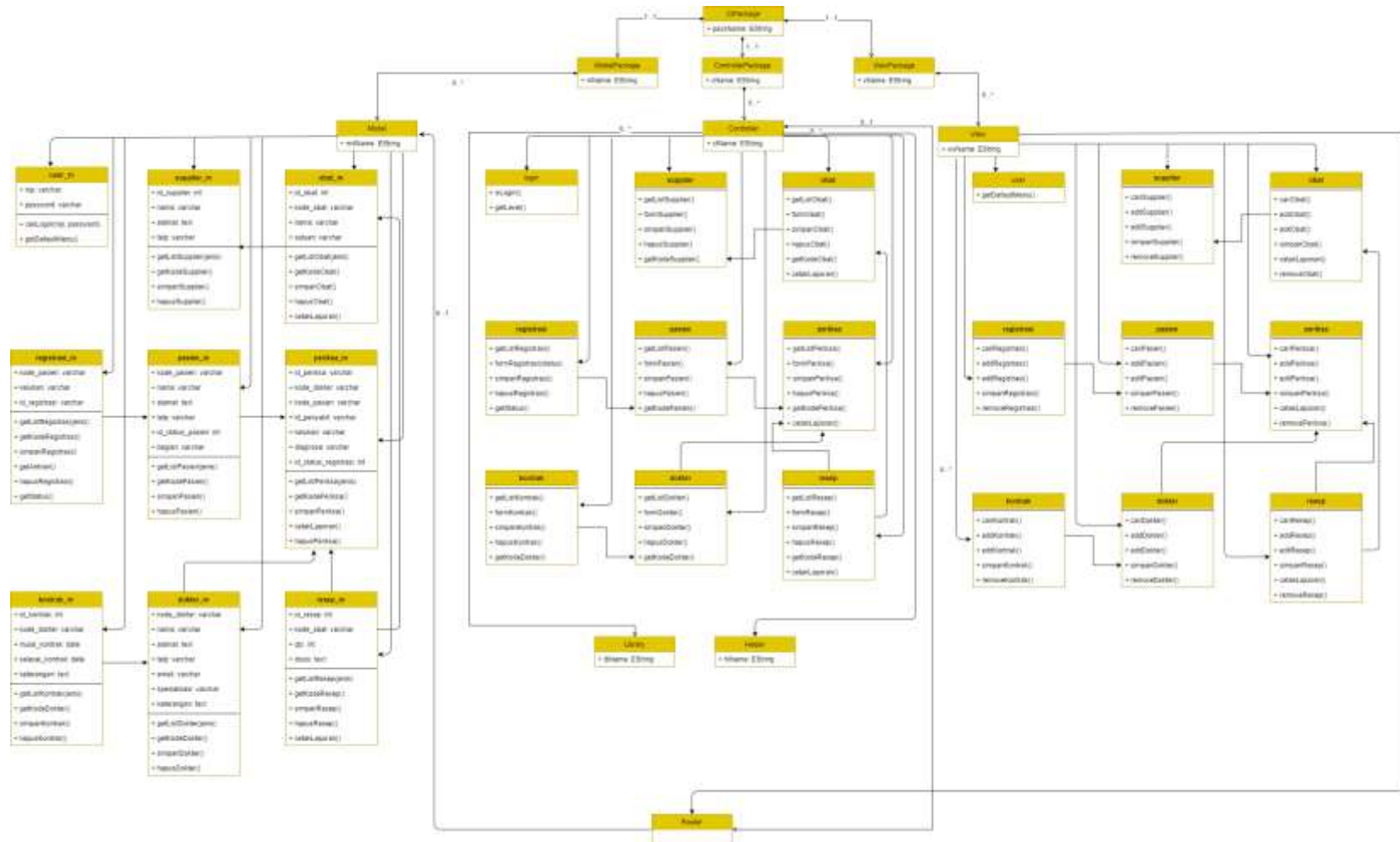
Selain mengelola data-data poliklinik, admin juga dapat mencetak data-data tersebut dengan langsung mencetak semua data atau mencetak data pada tanggal-tanggal tertentu. Kemudian sistem akan menampilkan *preview* laporan tersebut di *browser* dalam bentuk PDF.



Gambar 3. 19 Activity Diagram Cetak Laporan

### 3.1.2.5 Class Diagram

*Class diagram* berikut menampilkan hubungan antar *class* yang ada mencakup desain MVC (*model, view, controller*) sistem serta hubungannya dengan *class-class* yang lain. *Class* yang ada mencakup *login*, *pasien*, *stok obat*, *registrasi*, *pemeriksaan*, dan *resep*.

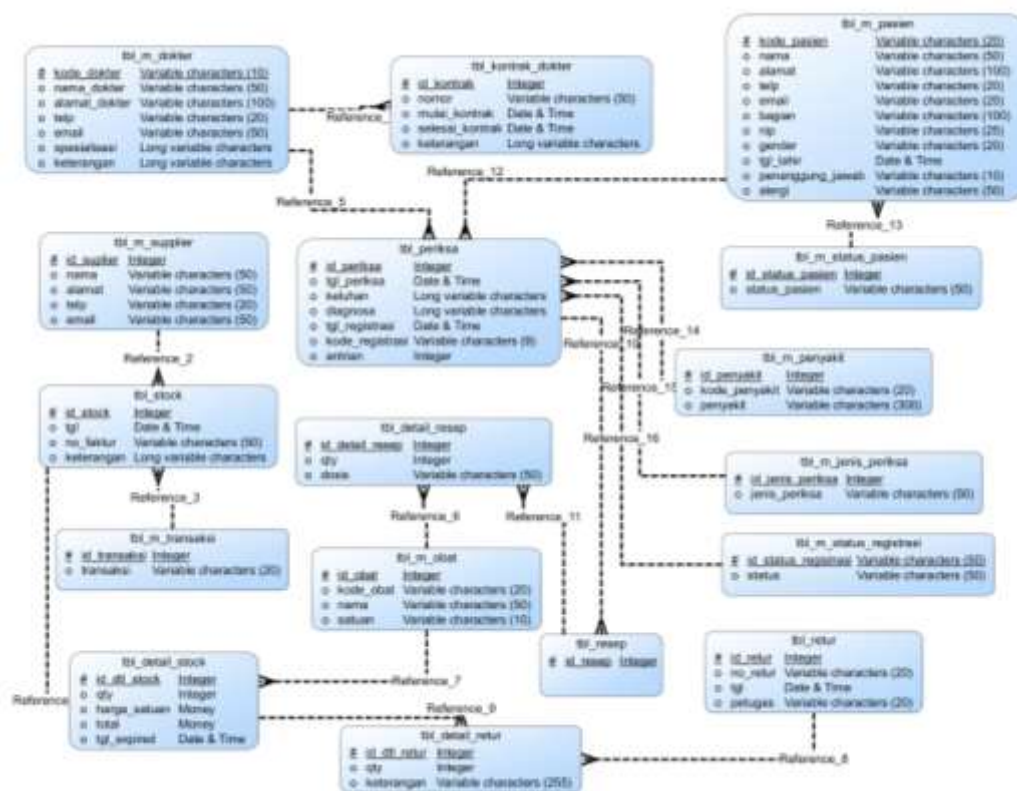


Gambar 3. 20 Class Diagram Sistem Informasi Poliklinik PDAM Kota Malan

### 3.1.2.3 Rancangan Basis Data

#### 3.1.2.3.1 Conceptual Data Model (CDM)

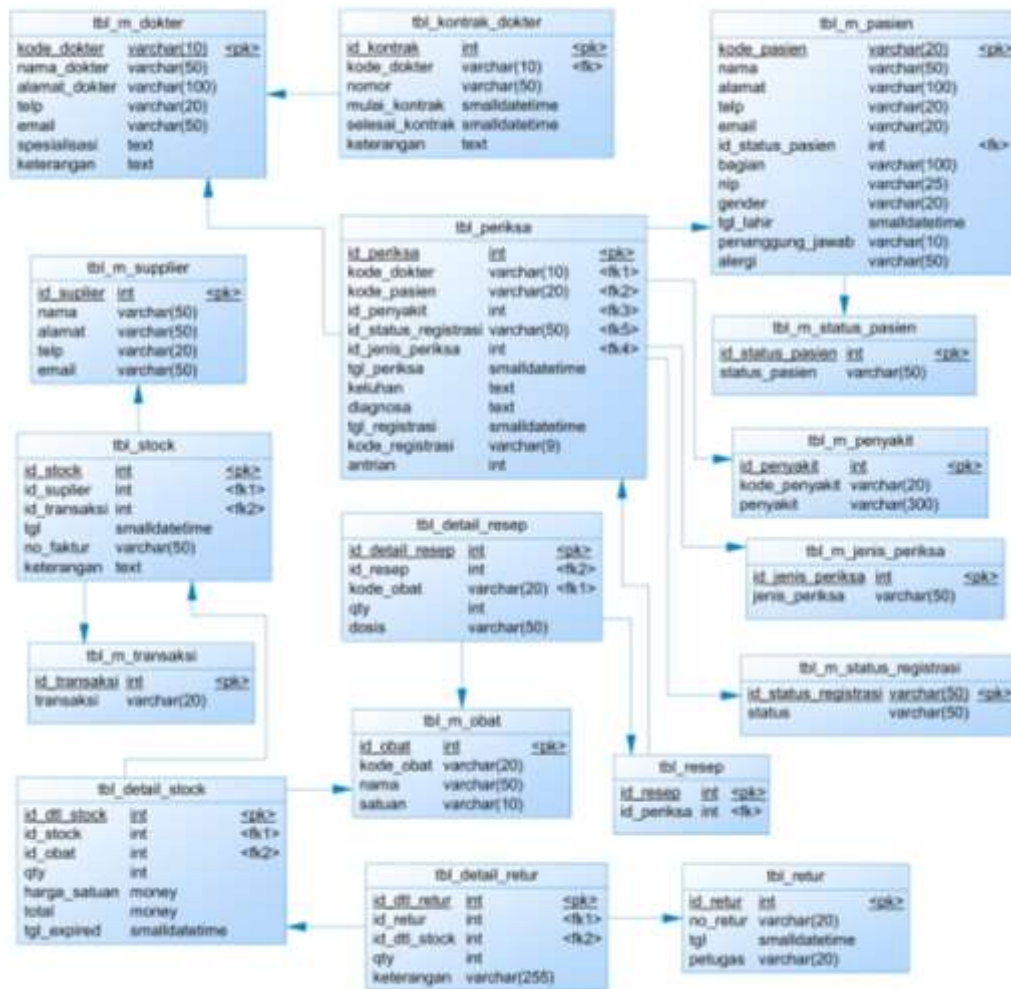
Diagram berikut merupakan Conceptual Data Model dari basis data Sistem Informasi Poliklinik PDAM Kota Malang yang kami terima dari pihak PDAM. Pembuatan CDM dimaksudkan untuk memodelkan struktur logis dari keseluruhan aplikasi meliputi struktur basis data yaitu arti, hubungan, dan batasan-batasan serta sebagai alat komunikasi antar pemakai basis data, desainer, dan analis.



Gambar 3. 21 *Conceptual Data Model*

#### 3.1.2.3.2 Physical Data Model (PDM)

Diagram berikut merupakan Physical Data Model dari basis data Sistem Informasi Poliklinik PDAM Kota Malang yang kami terima dari pihak PDAM. Pembuatan PDM dimaksudkan untuk menspesifikasikan implementasi secara fisik pada basis data.



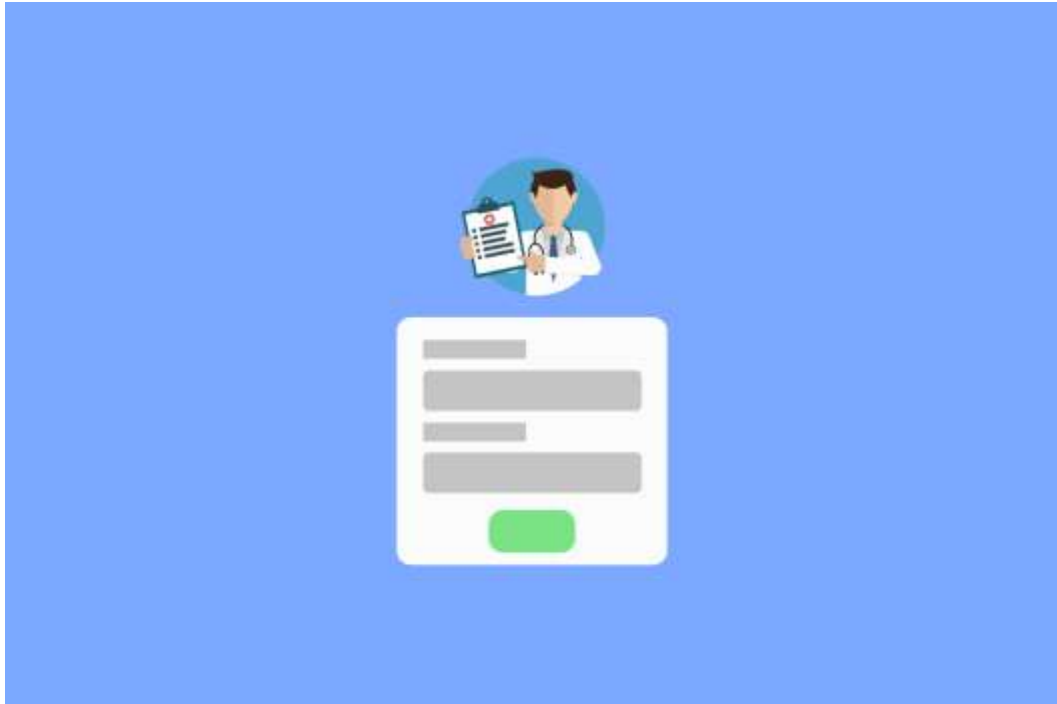
Gambar 3. 22 Physical Data Model

### 3.1.2.4 Rancangan Antarmuka Pengguna

#### 3.1.2.4.1 Halaman Login

Rancangan antarmuka halaman *login* dapat ditunjukkan gambar di bawah ini. Dimana pada halaman tersebut terdapat dua kolom isian untuk *username* dan *password* pengguna. Selain itu, juga terdapat *button login* memverifikasi inputan dari pengguna dan masuk ke halaman berikutnya.





Gambar 3. 23 Rancangan Halaman *Login*

#### 3.1.2.4.2 Halaman Tabel Data

Rancangan antarmuka halaman tabel data digunakan untuk menampilkan data dari kategori data master, data transaksi, dan data laporan. Pada halaman ini terdapat beberapa fitur pengolahan data seperti CRUD data dan *search* data. Dibagian samping halaman terdapat *sidebar* untuk mempermudah pengguna menuju data-data yang lain. Serta dibagian bawah halaman terdapat *pagination* untuk mempermudah pengguna melihat data di halaman selanjutnya.



Gambar 3. 24 Rancangan Halaman Tabel Data

#### 3.1.2.4.3 Halaman Form Tambah dan Edit Tabel Data

Rancangan antarmuka di bawah ini merupakan rancangan antarmuka untuk *form* edit dan tambah data. *Form* tersebut berisi beberapa kolom isian yang harus diisi oleh pengguna. Di bagian bawah *form* terdapat tombol simpan untuk menyimpan data yang diedit atau ditambah serta tombol batal jika tidak jadi mengedit atau menambah data.



Gambar 3. 25 Rancangan Halaman *Form* Tambah dan Edit Tabel Data

#### 3.1.2.4.4 Halaman Tabel Data Transaksi

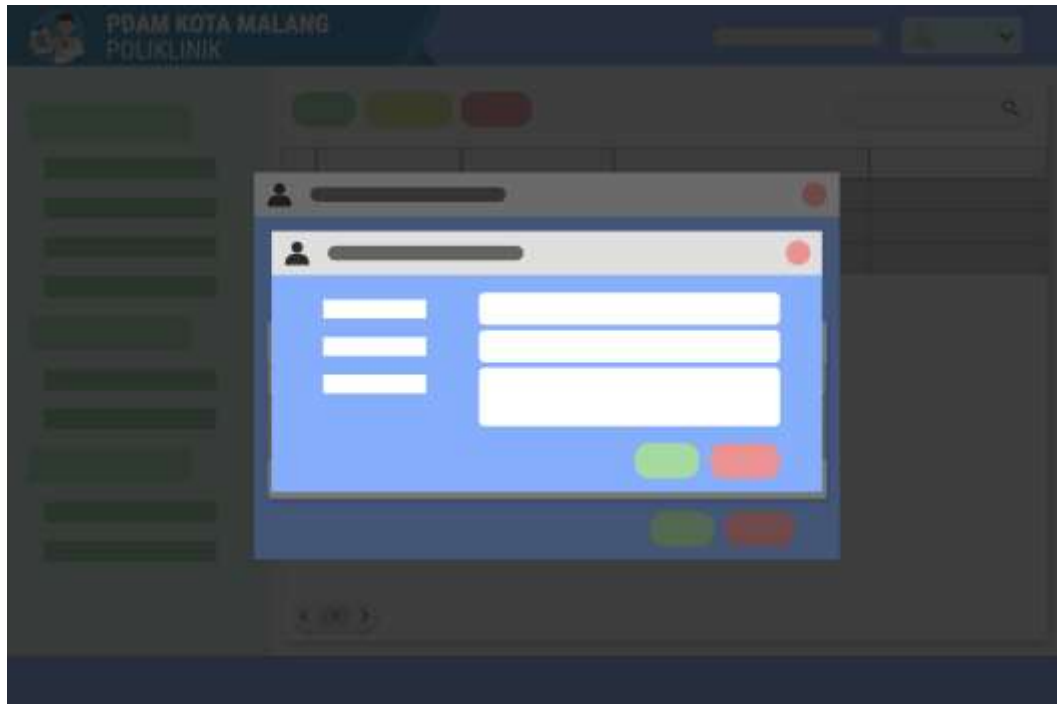
Beberapa tabel data transaksi seperti transaksi obat dan resep obat memiliki sub tabel di dalamnya. Desain dari sub tabel tersebut ditunjukkan pada gambar berikut. Tabel tersebut menampilkan detail data dari data yang dipilih. Pada tabel tersebut juga terdapat tombol untuk tambah, edit, dan hapus data.



Gambar 3. 26 Rancangan Halaman Tabel Data Transaksi

#### 3.1.2.4.5 Halaman Form Tambah dan Edit Tabel Data Transaksi

Pada sub tabel data transaksi apabila dilakukan proses edit dan tambah data, maka akan muncul *form* seperti gambar di bawah ini. *Form* tersebut berisi beberapa kolom isian yang harus diisi oleh pengguna. Di bagian bawah *form* terdapat tombol simpan untuk menyimpan data yang diedit atau ditambah serta tombol batal jika tidak jadi mengedit atau menambah data.



Gambar 3. 27 Rancangan Halaman *Form* Tambah dan Edit Tabel Data Transaksi

### 3.1.3 Evaluasi Prototipe

Evaluasi ini dilakukan oleh pelanggan, apakah prototipe yang sudah dibangun sesuai dengan keinginan pelanggan atau belum. Jika sudah selesai, maka langkah selanjutnya akan diambil. Namun jika tidak, prototipe direvisi dengan mengulang langkah-langkah sebelumnya.

### 3.1.4 Perbaikan dan Perbarui Prototipe

Perbaikan dan perbaruan prototipe dilakukan jika dalam proses evaluasi ditemukan hal-hal yang belum sesuai dengan keinginan pelanggan. Perbaikan akan terus dilakukan selama masih ada revisi dari pelanggan. Jika prototipe sudah dapat memenuhi keinginan pelanggan, maka dapat dilanjutkan dengan proses berikutnya yaitu pengkodean sistem.

### 3.1.5 Pengkodean Sistem

Dalam tahap ini, prototipe yang sudah disepakati diterjemahkan kedalam bahasa pemrograman yang sesuai dengan kebutuhan pengguna pada proses identifikasi. Pengkodean sistem mencakup pengkodean pada bagian *backend* dan *frontend*. Setelah proses pengkodean selesai, maka dapat dilanjutkan dengan proses pengujian sistem.

### **3.1.6 Pengujian Sistem**

Pada proses pengujian sistem kami melakukan dua pengujian yakni pengujian fungsionalitas dengan menggunakan *blackbox testing* serta pengujian terhadap pengguna menggunakan *User Acceptance Test (UAT)*. Pengujian fungsionalitas dilakukan untuk memastikan sistem dapat menangani semua masukan yang tidak sesuai dan memastikan pengguna hanya bisa memasukkan data yang benar ke sistem. Pengujian pengguna digunakan untuk menguji apakah sistem sudah sesuai dengan kebutuhan pengguna atau belum.

### **3.1.7 Pemeliharaan Sistem**

Sistem yang telah diuji dan diterima pelanggan siap untuk digunakan. Untuk keberlangsungan jalannya sistem maka dapat dilakukan pemeliharaan/*maintenance* secara berkala untuk memastikan sistem masih dapat berfungsi dengan baik.