

PERANCANGAN SISTEM POINT OF SALE TERINTEGRASI BERBASIS APLIKASI DESKTOP MENGGUNAKAN FLUTTER PADA APOTEK SF

Gevanno Pascal Yohanes¹, Retno Mumpuni^{2*}, Hazna At Thooriqoh³

^{1,2,3}*Informatika, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, Indonesia*

*Penulis korespondensi: retnomumpuni.if@upnjatim.ac.id

ABSTRAK

Apotek SF merupakan tempat menjual obat-obatan yang legal dan terpercaya di area Jawa Timur. Apotek SF menghadapi permasalahan terkait aplikasi *Point of Sale* (POS) yang digunakan saat ini karena belum terintegrasi antara proses pencatatan transaksi dengan proses pengelolaan stok obat secara menyeluruh. Oleh sebab itu, apotek SF beberapa kali mengalami masalah ketidakakuratan riwayat stok akibat dari sistem yang masih berjalan secara parsial. Penelitian ini bertujuan untuk memberi solusi terhadap masalah tersebut berupa perancangan sistem *Point of Sale* (POS) terintegrasi berbasis aplikasi *desktop* menggunakan *framework* Flutter yang dirancang khusus berdasarkan kebutuhan apotek SF. Penelitian ini menerapkan pendekatan *Object-Oriented Analysis and Design* (OOAD) yang berfokus pada analisis kebutuhan sistem dan perancangan sistem. Ruang lingkup penelitian ini hanya mencakup tahap perancangan sistem dan perancangan sistem ini menggunakan *Unified Modeling Language* (UML) sebagai bahasa pemodelan visual yang mencakup *Use Case Diagram*, *Activity Diagram*, *Sequence diagram*, dan *Class Diagram*. Sistem POS ini dirancang khusus untuk dua aktor pengguna saja, yaitu pemilik perusahaan sebagai admin dan apoteker sebagai karyawan dengan fitur utamanya meliputi pencatatan transaksi, kelola data obat, kelola catatan *restock*, *stock opname*, retur dan melihat laporan keuangan sederhana. Penelitian ini menghasilkan perancangan sistem dengan pemodelan UML sebagai landasan dalam proses implementasi yang akan datang dalam bentuk aplikasi *desktop* menggunakan *framework* Flutter.

Kata kunci: apotek, *point of sale*, aplikasi *desktop*, UML, Flutter.

1 PENDAHULUAN

Apotek merupakan tempat untuk masyarakat mendapatkan obat-obatan legal dan terpercaya yang dibutuhkan. Apotek memiliki peran penting pada sistem kesehatan nasional, karena apotek dalam skala apotek mandiri tersebar meluas, bahkan ada yang di tempat jarang penduduk sekalipun. Namun, apotek mandiri memiliki tantangan internal, yaitu pengelolaan stok obat secara teratur untuk menjaga kualitas obat-obatan yang hendak dijual. Apotek SF merupakan apotek studi kasus dari penelitian ini dengan skala apotek mandiri yang berdiri dan beroperasi di Jawa Timur, tepatnya berada di tengah-tengah pedesaan antara Kabupaten Mojokerto dengan Kabupaten Lamongan. Apotek SF menghadapi tantangan tersebut akibat kesalahan penerapan teknologi dalam mengelola stok yang dapat memicu kerugian finansial usaha (Rafif et al., 2025). Berdasarkan hasil wawancara dan observasi apotek SF telah menerapkan digitalisasi manajemen stok, namun alur manajemen stok masih berjalan secara parsial. Hal tersebut mengakibatkan beberapa proses bisnis masih dilakukan secara manual yang beberapa kali mengakibatkan masalah penumpukan stok karena obat tidak layak konsumsi (*deadstock*) dan kekosongan stok (*stockout*) (Masula & Winarno, 2025).

Upaya digitalisasi yang diterapkan pada Apotek SF saat ini berfokus pada penggunaan sistem *Point of Sale* (POS) untuk mencatat transaksi harian dan mengelola persediaan obat, meskipun kapasitas fiturnya masih terbatas. Secara mendasar, keberadaan sistem informasi hendaknya mampu menunjang seluruh aktivitas operasional dan manajerial organisasi melalui

penyatuan komponen manusia, perangkat keras, perangkat lunak, basis data, hingga prosedur kerja secara efisien (Nugraha & Suendri, 2024). Peran tersebut secara spesifik diakomodasi oleh sistem POS yang bertugas mengelola transaksi penjualan, menghitung pembayaran secara otomatis, mengatur persediaan barang, serta menyusun laporan berkala untuk mempercepat alur layanan sekaligus meminimalkan potensi kesalahan pencatatan (Apriyani et al., 2022).

Fitur terbatas pada aplikasi POS yang digunakan oleh apotek SF adalah hanya sebatas mengelola stok produk, mencatat *restock*, dan melihat laporan penjualan saja. Sementara itu, apotek SF memiliki 2 proses bisnis lainnya yang masih dilakukan secara manual di luar penggunaan aplikasi POS tersebut, antara lain proses *stock opname* yang masih dicatat pada grup WhatsApp, dan proses retur/pengembalian produk yang sudah dibeli oleh pelanggan. Hal tersebut membuat pengguna melakukan pembaharuan stok dengan beberapa aktivitas fiktif. Ketika melakukan *stock opname* dan ditemukan obat yang memiliki selisih, jika selisihnya lebih sedikit dari stok sistem yang dilakukan pengguna adalah mengurangi stok obat tersebut dengan cara membuat transaksi fiktif, jika selisihnya lebih banyak dari stok sistem yang dilakukan pengguna adalah menambah stok obat tersebut dengan cara membuat pencatatan *restock* fiktif. Ketika proses retur/pengembalian produk yang sudah dibeli oleh pembeli, pengguna mencatat melalui kertas catatan dan menambahkan stok obat yang dikembalikan dengan cara membuat pencatatan *restock* fiktif. Kondisi tersebut menyebabkan ketidakakuratan pencatatan stok, serta menyulitkan pegawai dalam memonitor stok. Hasil penelitian terdahulu menjelaskan bahwa fasilitas kesehatan dan apotek waralaba sering menanggung kerugian akibat *deadstock*, sehingga mengalami kerugian (Khairani et al., 2021). Masalah *stockout* pada konteks layanan kesehatan dapat menyebabkan turunnya loyalitas pelanggan karena apotek tidak dapat menyediakan obat-obatan yang diperlukan masyarakat (A. Nugraha, 2021).

Tujuan dari penelitian ini adalah dengan merancang sistem *Point of Sale* (POS) terintegrasi untuk memenuhi kebutuhan proses bisnis pada apotek SF. Sistem yang dirancang akan memiliki satu basis data terpusat untuk penerapan integrasi antara pencatatan transaksi dengan manajemen stok keseluruhan, antara lain kelola data obat, kelola pencatatan *restock*, kelola pencatatan retur obat, kelola pencatatan *stock opname*, dan melihat laporan keuangan. Dalam paper ini, fokus utama diarahkan pada tahap perancangan sistem dengan memanfaatkan *Unified Modeling Language* (UML) sebagai alat pemodelan. Beberapa diagram yang digunakan mencakup *Use Case Diagram*, *Activity Diagram*, *Sequence diagram*, dan *Class Diagram*. Sebagai standar yang diakui pada industri perangkat lunak, UML membantu memvisualisasikan arsitektur sistem secara terstruktur untuk menyampaikan konsep teknis kepada para pemangku kepentingan (*stakeholder*) dengan lebih mudah dan jelas (Narulita et al., 2024).

2 METODE

2.1 Metode Pengumpulan Data

Metode pengumpulan data merupakan proses untuk memperoleh data dan informasi yang dibutuhkan untuk menyelesaikan masalah pada penelitian. Penelitian ini menggunakan 2 metode pengumpulan data, yaitu melakukan observasi dan wawancara.

2.1.1 Observasi

Observasi merupakan proses mengumpulkan data secara langsung dengan mengamati situasi atau subjek di lapangan untuk menangkap fenomena yang terjadi apa adanya (Putri & Murhayati, 2025). Dalam penelitian ini dilakukan observasi untuk mengumpulkan informasi kegiatan dan proses alur transaksi di apotek SF dalam satu hari penuh. Kegiatan dari alur transaksi dimulai dari awal toko buka pada jam 7 pagi, alur melayani pelanggan yang hendak membeli obat/cek kesehatan, alur apoteker mencatat *restock*, alur apoteker melakukan *stock opname*, hingga diakhiri dengan pemilik perusahaan memeriksa ulang keuangan harian saat toko tutup pada jam 9 malam.

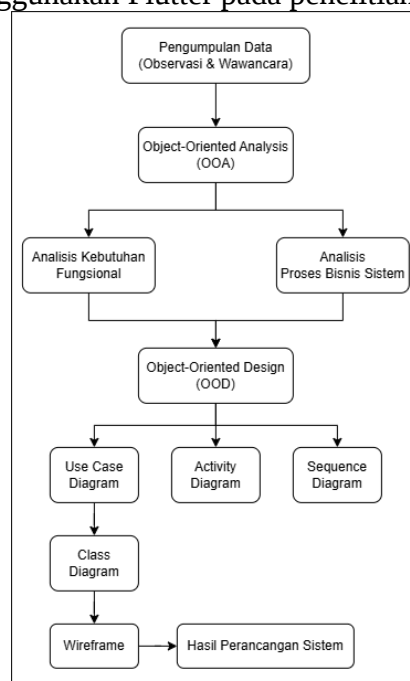
2.1.2 Wawancara

Wawancara merupakan proses tanya jawab antara peneliti dengan informan untuk menggali data secara mendalam dan menyeluruh terkait situasi/kondisi yang sedang diteliti (Rahmawati et al., 2025). Dalam penelitian ini dilakukan wawancara secara tatap muka dengan pemilik perusahaan dan salah satu apoteker/karyawan apotek SF untuk menggali kendala selama proses bisnis yang berlangsung menggunakan aplikasi *Point of Sale* (POS) yang masih berjalan secara parsial. Tujuan dari wawancara ini untuk membentuk analisis kebutuhan sistem yang akan dibuat.

2.2 Metode Object-Oriented Analysis and Design (OOAD)

Object-Oriented Analysis dan Design (OOAD) merupakan pendekatan dalam pengembangan perangkat lunak yang berorientasi objek dengan memisahkan proses menjadi dua tahapan utama, yaitu *Object-Oriented Analysis* (OOA) dan *Object-Oriented Design* (OOD). Pendekatan ini digunakan untuk menerjemahkan kebutuhan pengguna menjadi rancangan sistem yang terstruktur melalui identifikasi objek, hubungan antarkomponen, serta perilaku sistem sehingga menghasilkan cetak biru (*blueprint*) yang dapat dijadikan acuan pada tahap implementasi perangkat lunak (Mahfuzhi et al., 2025). Berbeda dengan metode pengembangan perangkat lunak yang berfokus pada implementasi dan perancangan sistem secara berorientasi objek sebelum memasuki tahap pembangunan perangkat lunak (Zulkarnain & Novita, 2023).

Pendekatan OOAD dipilih karena ruang lingkup penelitian ini dibatasi hingga tetap analisis kebutuhan dan perancangan sistem tanpa melakukan implementasi perangkat lunak. Hasil penelitian difokuskan pada penyusunan kebutuhan sistem, pemodelan menggunakan *Unified Modeling Language* (UML), serta perancangan wireframe sebagai rancangan awal antarmuka pengguna. Dengan demikian, hasil yang diperoleh berupa dokumen perancangan yang dapat digunakan sebagai landasan pada proses implementasi sistem *Point of Sale* (POS) berbasis aplikasi dekstop menggunakan Flutter pada penelitian selanjutnya.



Gambar 1. Tahapan *Object-Oriented Analysis and Design* (OOAD)

Berdasarkan gambar 1, penelitian ini diawali dengan proses pengumpulan data melalui observasi dan wawancara, kemudian dilanjutkan menggunakan pendekatan *Object-Oriented Analysis and Design* (OOAD) yang terdiri atas dua tahapan utama, yaitu *Object-Oriented*

Analysis (OOA) dan *Object-Oriented Design* (OOD). Tahap OOA berfokus pada analisis kebutuhan sistem berdasarkan hasil pengumpulan data, sedangkan tahap OOD bertujuan menerjemahkan kebutuhan tersebut ke dalam rancangan sistem berorientasi objek. Adapun uraian dari masing-masing tahapan dijelaskan sebagai berikut.

1. *Object-Oriented Analysis* (OOA)

Tahap *Object-Oriented Analysis* merupakan proses menganalisis kebutuhan sistem berdasarkan data yang telah diperoleh melalui observasi dan wawancara pada tahap pengumpulan data. Analisis dilakukan untuk mengidentifikasi kebutuhan fungsional dan kebutuhan nonfungsional yang harus dipenuhi oleh sistem *Point of Sale* (POS) sesuai dengan proses bisnis di Apotek SF (Mahfuzhi et al., 2025). Hasil dari tahapan ini berupa spesifikasi kebutuhan sistem yang menjadi dasar dalam proses perancangan.

2. *Object-Oriented Design* (OOD)

Tahap *Object-Oriented Design* merupakan proses menerjemahkan hasil analisis kebutuhan ke dalam rancangan sistem yang berorientasi objek. Perancangan dilakukan menggunakan bahasa pemodelan visual *Unified Modeling Language* (UML) yang meliputi *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram* untuk menggambarkan kebutuhan fungsional, alur proses, interaksi antarobjek, serta struktur kelas pada sistem (Puspaningrum et al., 2025). Selain pemodelan UML, penelitian ini juga menyusun rancangan antarmuka pengguna (*wireframe*) sebagai representasi awal tata letak halaman sistem sehingga dapat memberikan gambaran mengenai interaksi pengguna dengan sistem yang akan dikembangkan (Rachmatika et al., 2025). Hasil dari tahapan ini berupa dokumen perancangan sistem yang menjadi acuan pada proses implementasi perangkat lunak pada penelitian selanjutnya.

2.3 Pemodelan Perancangan

Dalam penelitian ini, perancangan sistem akan menggunakan bahasa pemodelan visual *Unified Modeling Language* (UML). UML berfungsi sebagai pemodelan perancangan sistem yang berorientasi objek yang digunakan untuk menggambarkan dan menyusun dokumentasi arsitektur perangkat lunak secara visual dengan berbagai jenis diagram (Haviluddin, 2016). Dengan memanfaatkan kumpulan diagram yang saling terhubung untuk menggambarkan perangkat lunak dari berbagai perspektif, mulai dari fungsionalitas sistem, interaksi antarobjek, dan arsitektur teknisnya (Setiaji & Sastra, 2021). UML menjadi bagian penting bagi analis dan pengembang, karena dari kebutuhan bisnis yang rumit dapat diterjemahkan menjadi sebuah cetak biru (*blueprint*) visual yang terstruktur serta mudah dipahami (Nurshadrina & Voutama, 2022). Pemodelan UML dalam penelitian ini meliputi *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*.

3 HASIL DAN PEMBAHASAN

3.1 Analisis Kebutuhan Sistem

Berdasarkan hasil observasi dan wawancara bersama pemilik perusahaan dan salah satu apoteker/karyawan apotek SF, diketahui bahwa apotek SF telah menggunakan aplikasi *Point of Sale* (POS) yang berlangganan. Namun, penggunaan aplikasi tersebut masih belum maksimal karena aplikasi POS tidak mendukung secara keseluruhan dari proses bisnis pada apotek SF. Sistem yang dirancang wajib memuat autentikasi pengguna dengan *role checking* untuk 2 aktor pengguna, yaitu pemilik perusahaan sebagai admin dan apoteker sebagai karyawan. Pembagian aktor pengguna berdasarkan kebutuhan fungsional dijelaskan pada tabel 1 di bawah ini.

Tabel 1. Pembagian aktor pengguna

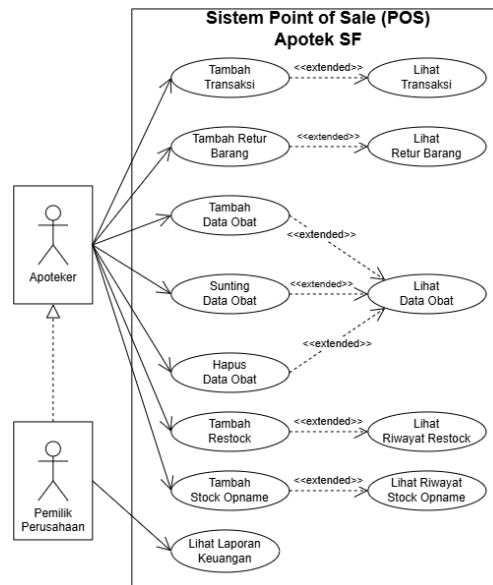
No.	Deskripsi Kebutuhan Fungsional	Aktor
1	Sistem memiliki fitur autentikasi (<i>login dan logout</i>).	Admin & Karyawan
2	Sistem dapat mengelola data obat yang dijual.	Admin & Karyawan
3	Sistem dapat menambah dan menyajikan data transaksi dengan pembaruan stok secara otomatis.	Admin & Karyawan
4	Sistem dapat menambah dan menyajikan data retur transaksi dengan pembaruan stok secara otomatis.	Admin & Karyawan
5	Sistem dapat menambah dan menyajikan data <i>restock</i> dengan pembaruan stok secara otomatis.	Admin & Karyawan
6	Sistem dapat menambah dan menyajikan data <i>stock opname</i> dengan pembaruan stok secara otomatis.	Admin & Karyawan
7	Sistem dapat menyajikan laporan keuangan sederhana.	Admin

Selain kebutuhan fungsional, adapun kebutuhan non fungsional yang dibutuhkan dalam pembuatan sistem ini, antara lain:

1. Aspek keamanan pada hak akses pengguna, karena sistem ini memiliki 2 aktor pengguna yang setiap aktor memiliki perbedaan hak akses, sehingga diperlukan keamanan hak akses pengguna berupa *role checking* untuk memisahkan aktor apa yang tidak dapat membuka fitur laporan.
2. Sistem berbasis aplikasi *desktop* pada sistem operasi Windows.
3. Sistem dapat digunakan dalam kondisi luring maupun daring. Dalam kasus ini, kondisi luring hanya untuk fitur tambah data transaksi saja. Aplikasi dirancang dengan fokus utama pada perangkat daring supaya aplikasi dapat bekerja dalam banyak perangkat yang berbeda (tidak terpaku pada 1 perangkat saja), dan dapat diakses setiap saat tanpa terbatas lokasi.
4. Sistem dapat berjalan dengan perangkat eksternal seperti *printer struk thermal* dan *cash drawer*.

3.2 Use Case Diagram

Use case diagram adalah salah satu diagram visual paling dasar dalam UML yang digunakan untuk menggambarkan batasan serta fungsi-fungsi utama suatu sistem informasi berdasarkan cara pengguna berinteraksi dengan sistem. Berikut disajikan *use case diagram* dari sistem *Point of Sale* (POS) pada gambar 2 di bawah ini.



Gambar 2. Use case diagram sistem

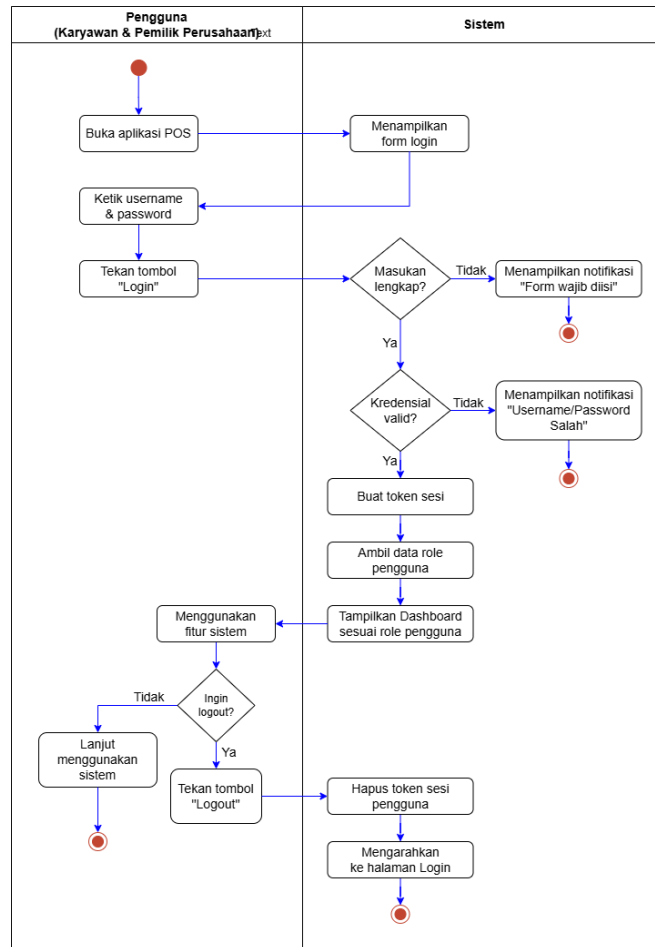
Berdasarkan gambar 2 tentang *use case diagram* tersebut, diketahui bahwa sistem POS yang dirancang memiliki dua aktor pengguna. Pengguna dengan aktor apoteker memiliki akses dasar pada proses bisnis di apotek SF, mulai dari menambah dan melihat data transaksi, menambah dan melihat data retur, mengelola data obat, menambah dan melihat data *restock*, serta menambah dan melihat riwayat *stock opname*. Pengguna dengan aktor pemilik perusahaan memiliki seluruh akses pada proses bisnis di apotek SF dengan rincian sama seperti apoteker dan dapat melihat laporan keuangan.

3.3 Activity diagram

Pada *activity diagram*, alur kerja (*workflow*) atau rangkaian proses aktivitas pengguna pada sistem informasi divisualisasikan secara runtut dan jelas dari awal hingga akhir. Berikut diuraikan beberapa *activity diagram* yang menjelaskan alur kerja utama pada sistem POS yang dirancang.

1. Autentikasi

Proses awal untuk mengakses seluruh fitur pada sistem POS adalah dengan melakukan autentikasi *login*. Berikut disajikan *activity diagram* dari autentikasi pada sistem POS pada gambar 3 di bawah ini.

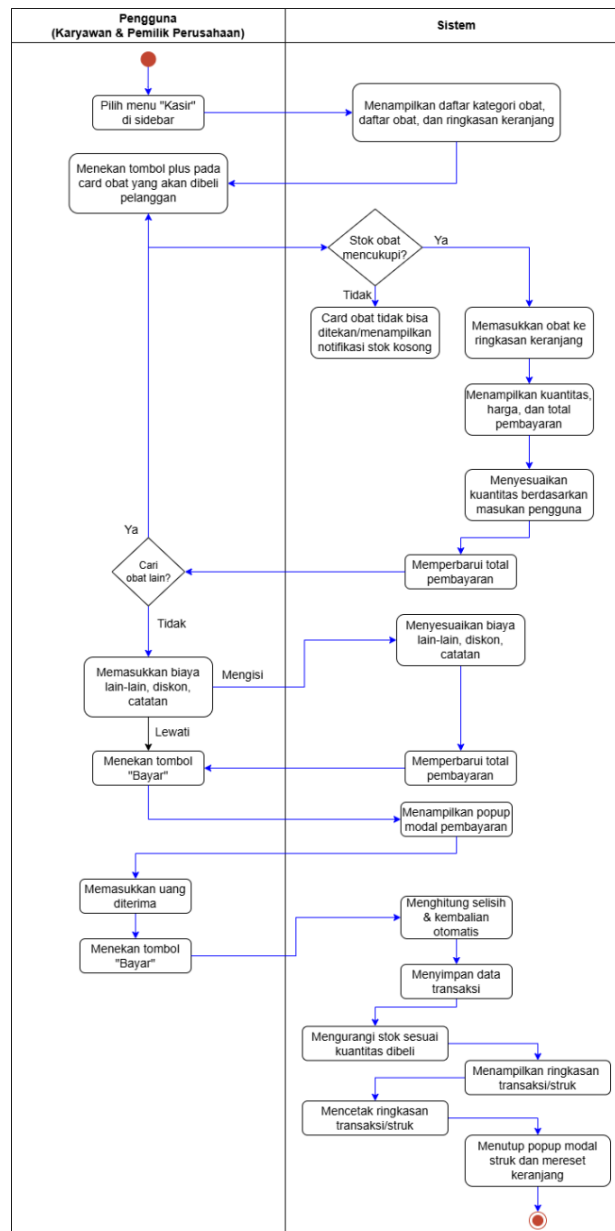


Gambar 3. Activity diagram autentikasi

Alur dimulai dengan proses validasi kredensial pengguna, jika kredensial dinyatakan valid, API akan membuat token sesi untuk disimpan pada aplikasi *desktop* dan memeriksa *role* akun. Setelah diperiksa, sistem akan mengarah ke halaman *dashboard* dan akan menampilkan menu apa saja yang dapat dibuka berdasarkan *role* akun.

2. Transaksi

Selanjutnya masuk pada proses bisnis pencatatan transaksi. Berikut disajikan *activity diagram* dari pencatatan transaksi pada sistem POS pada gambar 4 di bawah ini.

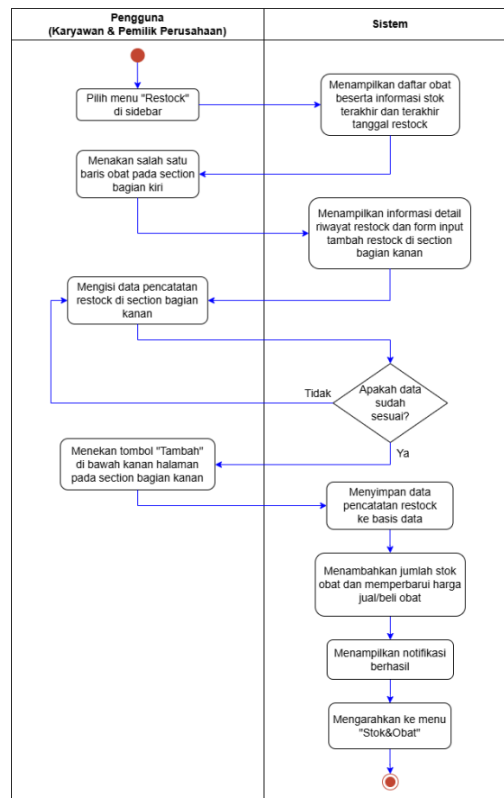


Gambar 4. Activity diagram pencatatan transaksi

Alur dimulai dengan pengguna masuk ke halaman kasir, lalu program menampilkan seluruh data obat beserta informasi stok yang tersimpan di basis data. Sistem secara otomatis akan membuat tampilan tombol + tidak bisa ditekan jika stok obat tidak tersedia. Jika stok tersedia, pengguna bisa memasukkan obat ke keranjang terlebih dahulu untuk mengatur biaya lain-lain dan menambah catatan jika diperlukan. Jika semua obat yang diperlukan sudah masuk keranjang, pengguna melanjutkan proses transaksi dengan memasukkan jumlah nominal uang yang dibayar pelanggan. Sistem secara otomatis akan membantu pengguna dalam menghitung uang kembalian. Proses terakhir sistem akan mengurangi jumlah stok obat yang dibeli dan mencetak struk untuk pelanggan.

3. Restock

Selanjutnya masuk pada proses bisnis pencatatan *restock*. Berikut disajikan *activity diagram* dari pencatatan *restock* pada sistem POS pada gambar 5 di bawah ini.

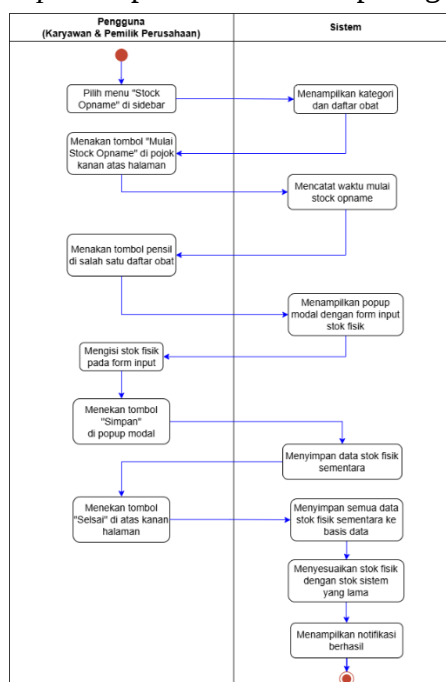


Gambar 5. Activity diagram pencatatan restock

Alur dimulai dengan pengguna masuk ke halaman *restock*, lalu program menampilkan daftar obat dan pengguna dapat memilih salah satu obat yang hendak dicatat. Lalu sistem menampilkan informasi riwayat *restock* dan pengguna dapat langsung mencatat *restock* obat tersebut. Proses terakhir sistem akan menambah jumlah stok obat sesuai masukan pengguna.

4. Stock opname

Selanjutnya masuk pada proses bisnis pencatatan *stock opname*. Berikut disajikan *activity diagram* dari pencatatan *stock opname* pada sistem POS pada gambar 6 di bawah ini.



Gambar 6. Activity diagram pencatatan stock opname

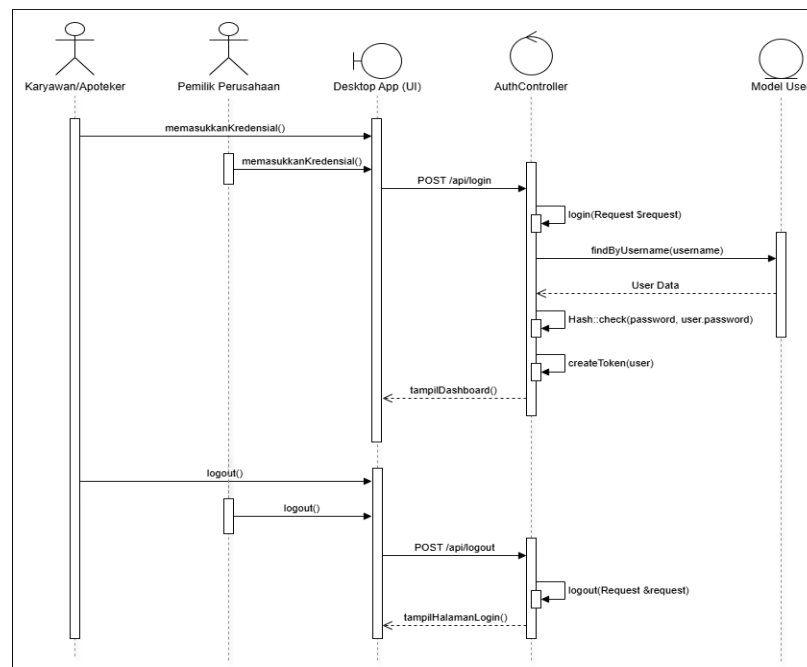
Alur dimulai dengan pengguna masuk ke halaman *stock opname*, lalu program menampilkan kategori dan daftar obat, lalu fitur *stock opname* akan bekerja ketika pengguna menekan tombol “Mulai *Stock opname*”, kemudian sistem secara otomatis akan mengubah tombol pensil menjadi bisa ditekan (karena sudah dalam kondisi *stock opname*), dan ketika pengguna menekan tombol pensil, sistem menampilkan modal dialog untuk pengguna dapat mengisi stok nyata di lapangan, jika tidak ada selisih maka pengguna dapat menyelesaikan kondisi *stock opname* dengan menekan tombol “Selesai” dan sistem akan secara otomatis memproses selisih stok yang dicatat.

3.4 Sequence diagram

Sequence diagram merupakan visualisasi dari interaksi dan pertukaran pesan antar objek di dalam suatu sistem berdasarkan urutan waktu eksekusinya. Berikut diuraikan:

1. Autentikasi

Berdasarkan *activity diagram* di atas, proses awal pada sistem POS adalah autentikasi *login*. Berikut disajikan *sequence diagram* dari autentikasi pada sistem POS pada gambar 7 di bawah ini.

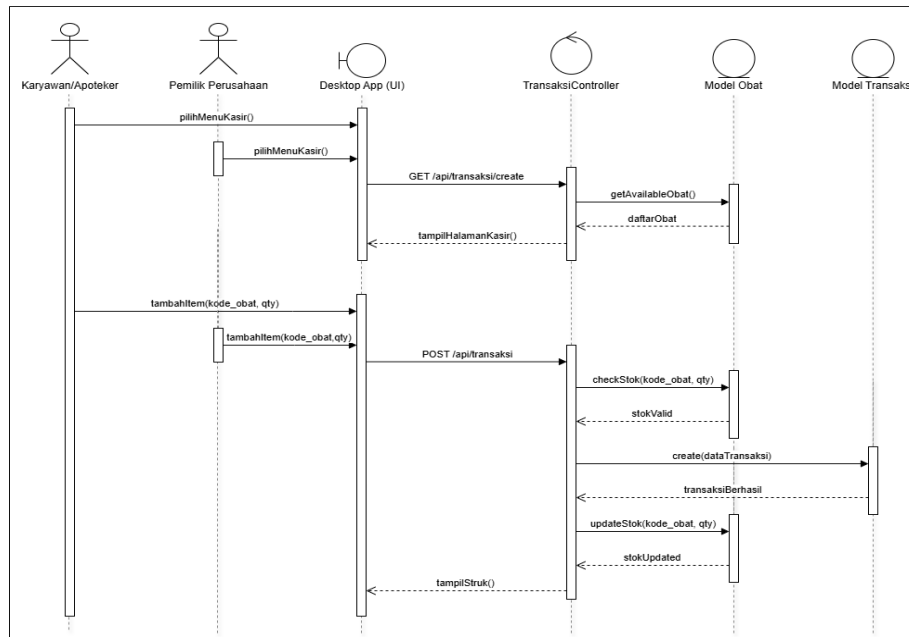


Gambar 7. Sequence diagram autentikasi

Mekanisme autentikasi sistem mencakup prosedur *login* dan *logout* bagi pengguna melalui antarmuka *desktop* yang terintegrasi dengan AuthController dan Model User. Pada proses *login*, kredensial dikirimkan permintaan pada API dengan metode POST untuk divalidasi melalui pencocokan data serta verifikasi *hash* kata sandi sebelum sistem menertibkan token akses ke halaman utama. Sementara itu, prosedur *logout* dijalankan dengan mengirimkan permintaan serupa untuk mengakhiri sesi aktif dan mengembalikan antarmuka ke halaman awal secara otomatis.

2. Transaksi

Selanjutnya masuk pada proses bisnis pencatatan transaksi. Berikut disajikan *sequence diagram* dari pencatatan transaksi pada sistem POS pada gambar 8 di bawah ini.

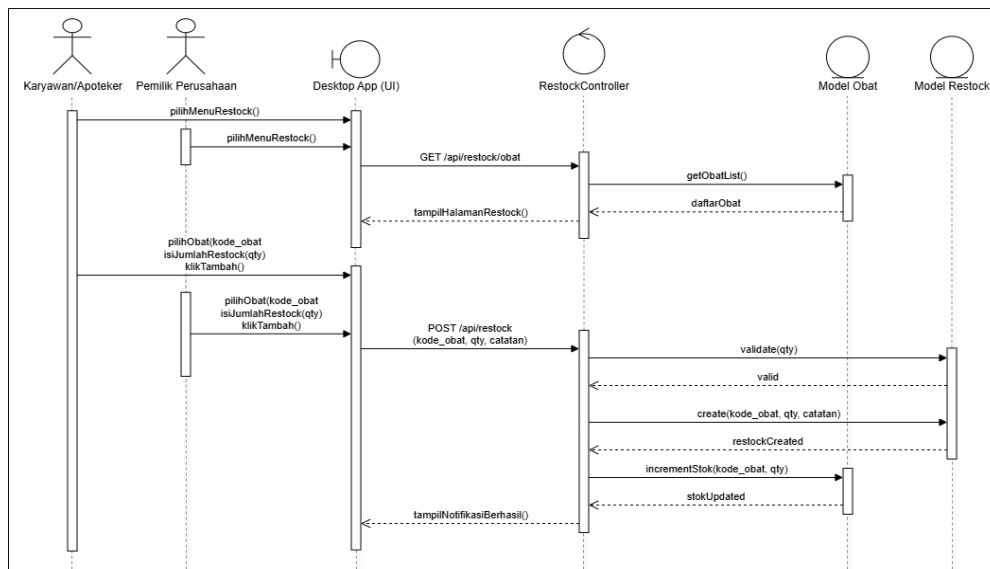


Gambar 8. Sequence diagram pencatatan transaksi

Alur pencatatan transaksi melibatkan interaksi antara antarmuka *desktop*, TransaksiController, serta Model Obat dan Model Transaksi. Proses diawali dengan pemuatan daftar obat yang tersedia melalui permintaan pada API dengan metode GET saat menu kasir dibuka. Kemudian, ketika data transaksi dikirim dengan metode POST, pengontrol akan memvalidasi kuantitas stok, menyimpan catatan penjualan baru, sekaligus memperbarui sisa stok obat sebelum akhirnya menampilkan struk belanja pada layar.

3. Restock

Selanjutnya masuk pada proses bisnis pencatatan transaksi. Berikut disajikan *sequence diagram* dari pencatatan transaksi pada sistem POS pada gambar 9 di bawah ini.



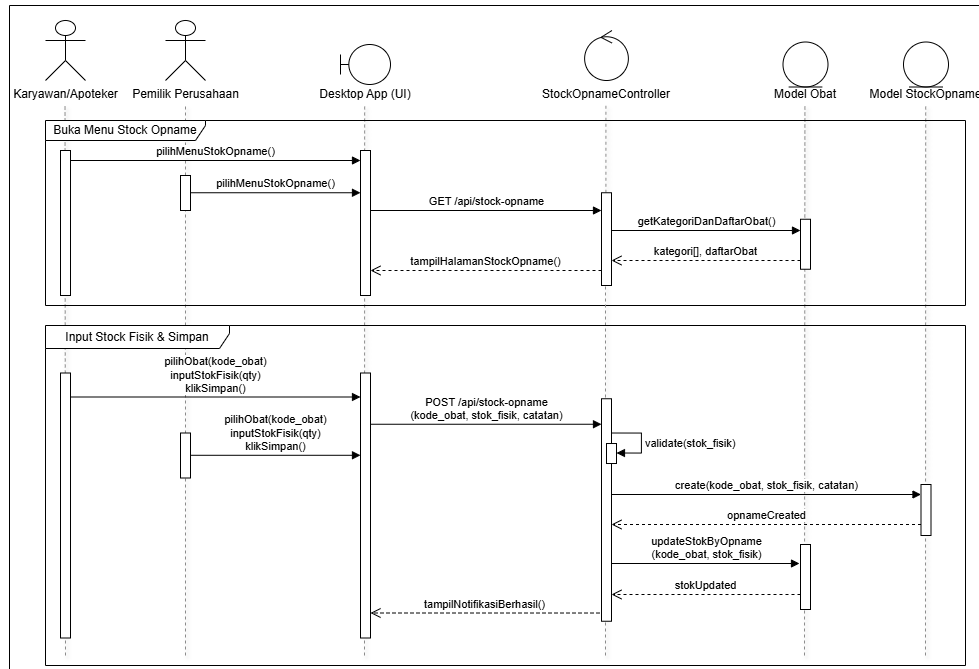
Gambar 9. Sequence diagram pencatatan restock

Alur *restock* obat melibatkan interaksi antara antarmuka *desktop*, RestockController, serta Model Obat dan Model Restock. Proses diawali dengan pemuatan daftar obat melalui permintaan pada API dengan metode GET saat menu *restock* dibuka. Kemudian, ketika data pasokan dikirim melalui metode POST, pengontrol akan memvalidasi data, menyimpan

pencatatan *restock* baru, dan menambah jumlah stok obat sebelum akhirnya menampilkan notifikasi keberhasilan pada layar.

4. *Stock opname*

Selanjutnya masuk pada proses bisnis pencatatan transaksi. Berikut disajikan *sequence diagram* dari pencatatan transaksi pada sistem POS pada gambar 10 di bawah ini.

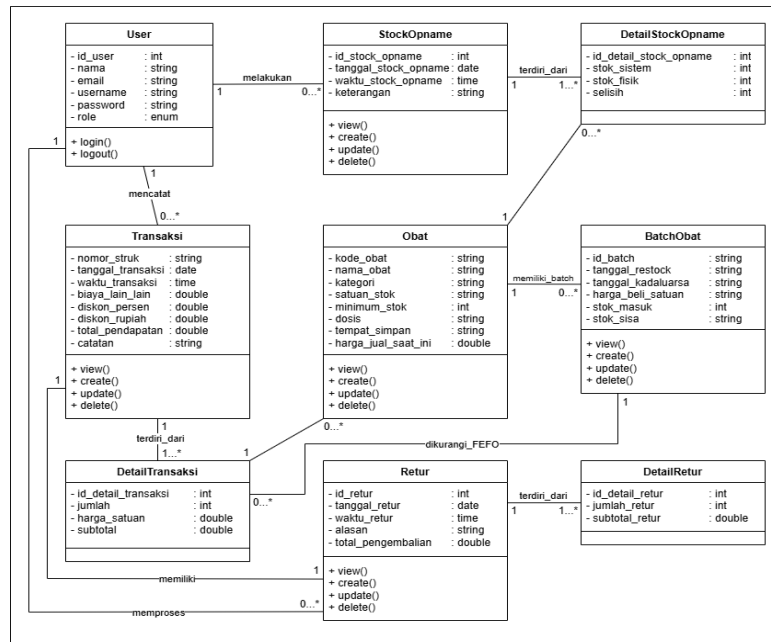


Gambar 10. *Sequence diagram* pencatatan stock opname

Proses *stock opname* diawali dengan pemuatan kategori dan daftar obat melalui permintaan pada API dengan metode GET pada antarmuka *desktop*. Ketika data stok fisik dikirim melalui metode POST, *StockOpnameController* akan memvalidasi input, menyimpan rekaman baru pada *Model StockOpname*, dan memperbarui stok obat pada *Model Obat* sebelum akhirnya menampilkan notifikasi keberhasilan.

3.5 *Class Diagram*

Class diagram menggambarkan arsitektur statis sistem perangkat lunak berupa gambaran entitas kelas, atribut, metode (*method*), dan relasi antar objek. Berikut disajikan *class diagram* dari sistem *Point of Sale* (POS) pada gambar 11 di bawah ini.



Gambar 11. Class diagram sistem POS

Class diagram ini dirancang berdasarkan hasil analisis kebutuhan sistem dan berfungsi untuk menggambarkan entitas nyata yang berjalan di apotek SF. Dalam mendukung fungsional sistem *Point of Sale* (POS) yang akan dibuat, struktur data terdiri dari beberapa kelas utama, meliputi User, Obat, BatchObat, Transaksi, DetailTransaksi, Retur, DetailRetur, StockOpname, dan DetailStockOpname. Seluruh kelas-kelas ini saling berintegrasi untuk menangani pencatatan transaksi, manajemen stok, proses *restock*, dan proses retur. Setiap relasi antar kelas juga dilengkapi dengan aturan kardinalitas untuk membatasi jumlah relasi objek sesuai kebutuhan di lapangan.

3.6 Wireframe

Tahap perancangan pada bagian antarmuka pengguna diawali dengan pembuatan *wireframe* untuk menyajikan sketsa tata letak antarmuka pengguna sebelum melangkah ke tahap penulisan kode. *Wireframe* ini dirancang dengan referensi dari *user interface* (UI) pada sistem yang lama, supaya pengguna nantinya dapat lebih mudah mempelajari tata letak tampilan sistem yang baru karena sudah terbiasa dengan tata letak tampilan sistem yang lama. Berikut diuraikan masing-masing desain *wireframe* pada beberapa halaman utama sistem *Point of Sale* (POS) untuk apotek SF.

1. Halaman Login

Apotek Satria Farma Mojokerto

Username:

Password: ☐

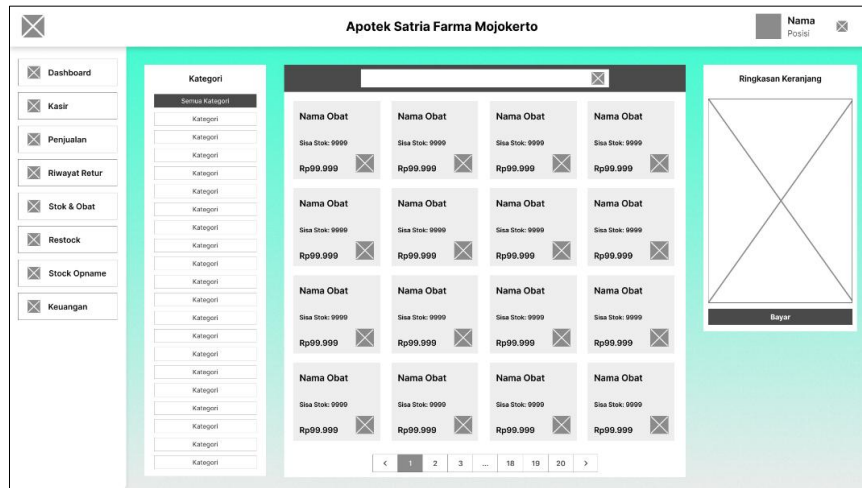
☒ Ingat Saya

LOGIN

Gambar 12. Wireframe halaman login

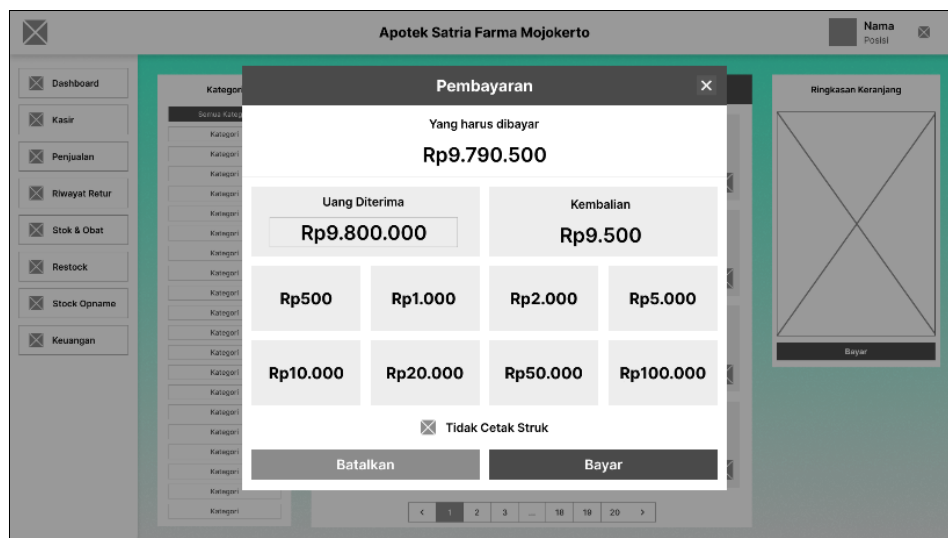
Halaman *login* hanya memuat kolom input *username* dan *password* dengan tambahan fungsional “Ingat Saya” berbentuk *radio button* untuk menyimpan kredensial yang berhasil *login*. Halaman ini didesain sederhana dan bersih untuk mengurangi pemborosan saat proses *slicing UI*. Halaman ini menjadi halaman awal sebelum pengguna dapat mengakses fitur utama aplikasi.

2. Halaman Kasir



Gambar 13. Wireframe halaman kasir

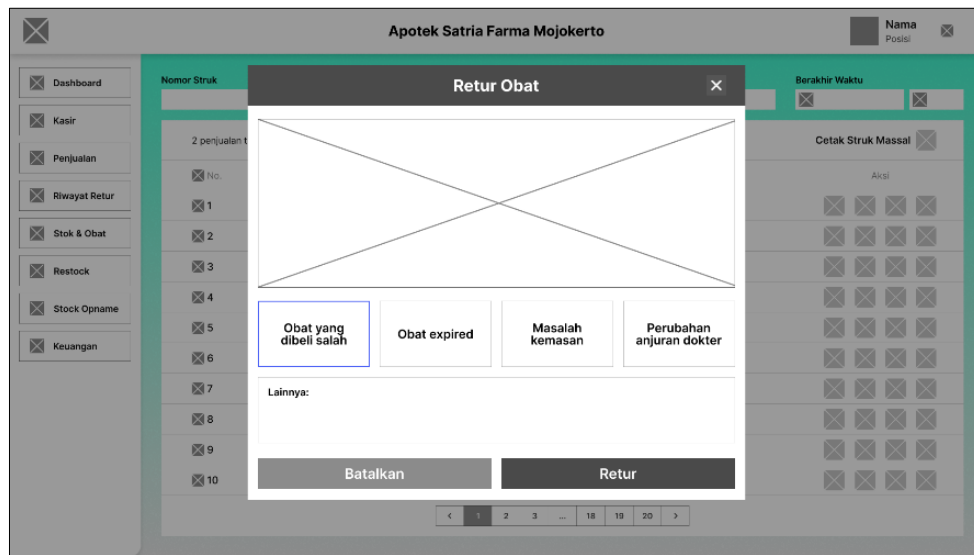
Halaman kasir memuat daftar kategori pada panel sisi kiri, memuat daftar obat pada panel sisi tengah, dan memuat ringkasan keranjang pada panel sisi kanan. Sistem daftar obat berbentuk *pagination*, dan bentuk daftar obat adalah kartu data obat yang menampilkan nama obat, sisa stok, dan harga jual obat, serta terdapat tombol untuk menambahkan ke keranjang. Penggunaan *pagination* untuk mengurangi *buffer* yang terlalu lama saat mengambil data obat yang berjumlah ribuan.



Gambar 14. Wireframe dialog pembayaran

Proses selanjutnya pada halaman kasir adalah muncul dialog pembayaran ketika pengguna menekan tombol “Bayar”. Dialog ini bertujuan untuk membantu pengguna dalam menyiapkan uang kembalian kepada pengguna dan memilih opsi jika pengguna tidak ingin mencetak struk transaksi.

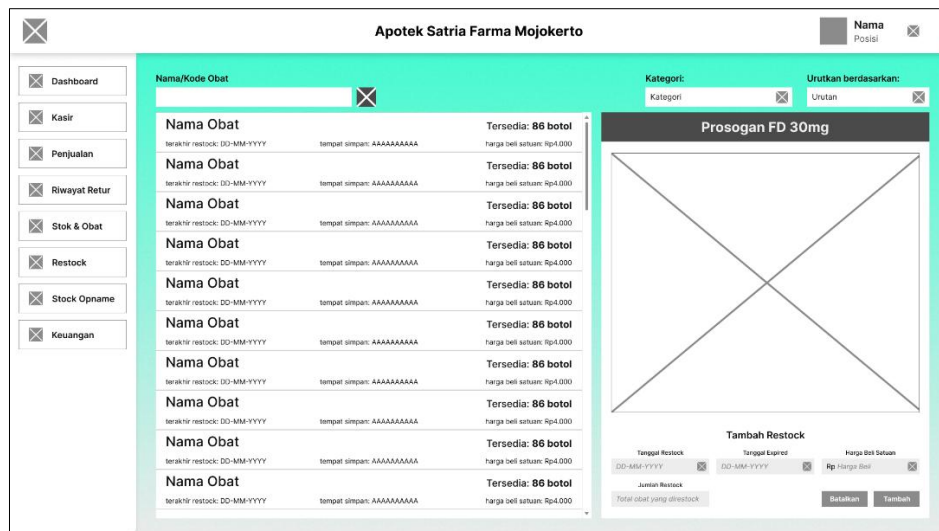
3. Halaman Retur



Gambar 15. Wireframe dialog retur

Dialog retur merupakan proses lanjutan dari halaman penjualan. Dialog ini muncul ketika pengguna menekan tombol retur pada kolom aksi di salah satu transaksi. Dialog ini bertujuan untuk mencatat retur obat yang dilakukan pelanggan dengan memilih alasan retur.

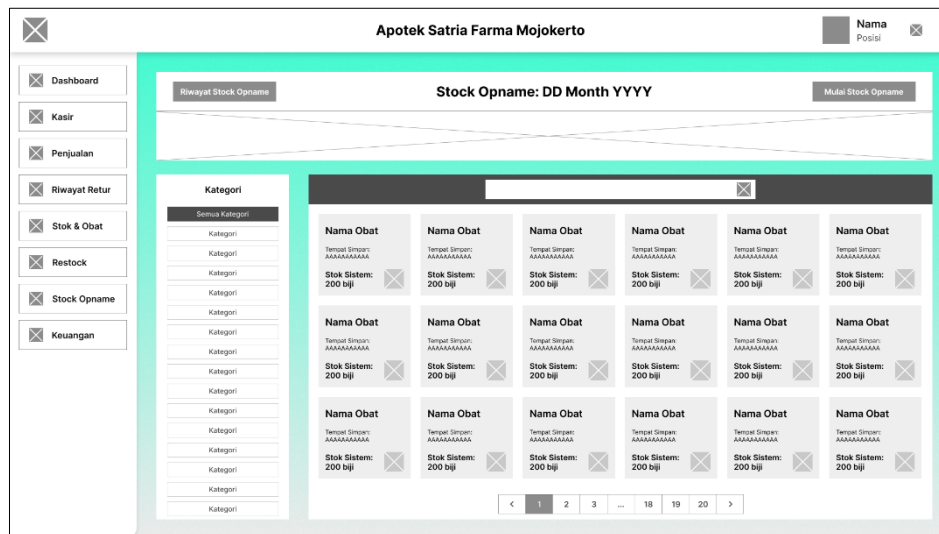
4. Halaman Restock



Gambar 16. Wireframe halaman restock

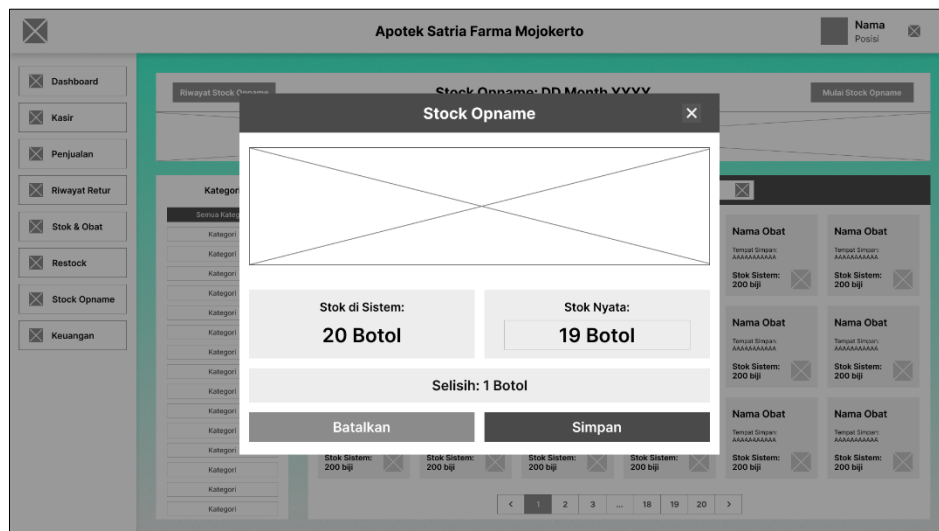
Halaman *restock* memuat daftar obat pada panel kiri, dan memuat data riwayat *restock* serta kolom input untuk tambah/sunting data *restock* pada panel kanan. Sistem daftar obat bukan berbentuk *pagination* karena data obat bukan berbentuk kartu melainkan dalam bentuk *list*.

5. Halaman *Stock opname*



Gambar 17. Wireframe halaman *stock opname*

Halaman *stock opname* memuat informasi waktu *stock opname* dan petunjuk penggunaan fitur pada halaman *stock opname* pada bagian panel atas, memuat daftar kategori pada bagian panel kiri, dan memuat daftar obat pada panel kanan. Kartu data obat ini memuat informasi nama obat, tempat simpan, stok pada sistem tercatat berapa, dan tombol untuk mencatat *stock opname*. Sistem daftar obat berbentuk *pagination* untuk mengurangi *buffer* dalam mengumpulkan daftar obat yang berjumlah ribuan.



Gambar 18. Wireframe dialog *stock opname*

Proses selanjutnya pada halaman *stock opname* adalah muncul dialog *stock opname* ketika pengguna menekan tombol “*Edit*”. Dialog ini bertujuan untuk mencatat selisih stok sistem dengan stok di lapangan pada obat yang dipilih.

3.7 Spesifikasi Infrastruktur

Tahap perancangan infrastruktur merupakan bagian penting sebelum masuk tahap pembuatan purwarupa. Dukungan infrastruktur juga perlu diperhitungkan mengingat sistem yang dirancang merupakan sistem yang berjalan secara daring, dan jika tidak diperhitungkan secara matang mengakibatkan performa sistem mengalami bottleneck karena ketidakmampuan peladen dalam memproses kueri dan permintaan aplikasi.

3.7.1. Spesifikasi Perangkat Keras (*Hardware*)

Pada penelitian ini terdapat 2 jenis perangkat keras yang dibutuhkan, antara lain perangkat keras untuk menjalankan sistem *Point of Sale* (POS) dan perangkat keras pendukung sistem POS. Berikut disajikan spesifikasi perangkat keras yang dibutuhkan pada tabel 2 di bawah ini.

Tabel 2. Spesifikasi perangkat keras

Komponen	Spesifikasi Minimum	Spesifikasi Rekomendasi
<i>Processor</i>	Intel Core i3/AMD Ryzen 3	Intel Core i5/AMD Ryzen 5
Memori (RAM)	4GB	8GB
Penyimpanan	SSD 128GB (diperlukan < 1GB)	NVMe SSD 256GB (diperlukan < 1GB)
Sistem Operasi	Windows 10 (64-bit)	Windows 11 (64-bit)

Spesifikasi perangkat keras tersebut memiliki spesifikasi yang tidak terlalu tinggi, karena sistem POS ini dirancang untuk dapat diakses diberbagai jenis spesifikasi perangkat laptop/PC dengan sistem operasi pada perangkat keras hanya untuk sistem operasi Windows saja. Selain itu, terdapat daftar perangkat pendukung yang diperlukan untuk mendukung penggunaan sistem POS yang disajikan pada tabel 3 di bawah ini.

Tabel 3. Perangkat pendukung operasional

Jenis Perangkat	Merk dan Tipe	Keterangan
<i>Cash Drawer</i>	Sharkpos CD-339	Digunakan untuk penyimpanan uang tunai.
<i>Thermal Printer</i>	IWare (Tipe Discontinued)	Digunakan untuk pencetakan struk belanja.

Merk dan tipe pada perangkat pendukung tersebut disesuaikan berdasarkan kepemilikan pada apotek SF. Jadi, sistem POS ini akan dibuat untuk dapat mendukung berbagai merk dan tipe *cash drawer* serta *thermal printer*, atau bersifat *universal*.

3.7.2. Spesifikasi Peladen (*Server-Side*)

Sistem *Point of Sale* (POS) yang dirancang dalam penelitian ini berbasis aplikasi *desktop* menggunakan Flutter dan dapat dijalankan secara daring, sehingga pada perancangan sistem POS ini diperlukan peladen untuk menyimpan seluruh data pada sistem POS supaya basis datanya terpusat. Berikut disajikan spesifikasi peladen yang dibutuhkan pada tabel 4 di bawah ini.

Tabel 4. Spesifikasi peladen

Kategori	Spesifikasi Teknis	Fungsi Utama
Layanan <i>Hosting</i>	<i>Business Hosting - Advanced</i> (ArenhostID)	Menjalankan API Laravel 11 dan basis data MySQL pusat.
Kapasitas SSD	<i>Unlimited SSD NVMe Storage</i>	Menampung API Laravel 11 tanpa batasan kapasitas.
Kapasitas Basis Data	<i>Unlimited MySQL Database</i>	Menampung data transaksi yang dinamis dan terpusat.
Sertifikasi Keamanan	<i>Standard SSL Certificate</i>	Menjamin keamanan transmisi data antara aplikasi & server.

Spesifikasi peladen disesuaikan dengan besarnya data transaksi dan data obat-obatan yang tersedia pada sistem POS lama. Diketahui bahwa sistem POS lama memiliki data keseluruhan lebih dari 400.000 data dari tahun 2022 hingga tahun 2025, sehingga peladen yang digunakan juga harus memiliki spesifikasi yang tinggi. Penggunaan sertifikasi keamanan juga perlu diperhatikan untuk mencegah risiko kebocoran data maupun kebobolan pada sistem

Application Programming Interface (API). Dengan spesifikasi tersebut, penulis berlangganan sebuah *shared hosting* untuk menghemat biaya sewa.

4 KESIMPULAN

Penelitian ini berhasil menyusun perancangan sistem *Point of Sale (POS)* untuk apotek SF menggunakan pemodelan *Unified Modeling Language (UML)*. Sistem yang dirancang menjawab atas permasalahan apotek SF mengenai sistem POS yang berjalan secara parsial dengan memberikan solusi sebuah perancangan sistem POS yang terintegrasi antara pencatatan transaksi, pengelolaan data obat, pencatatan *restock*, pencatatan retur, pencatatan *stock opname*, dan pelaporan keuangan secara sederhana. Hasil perancangan sistem ini menjadi desain dasar untuk melanjutkan pada tahap implementasi sistem POS yang terintegrasi berbasis aplikasi *desktop* menggunakan Flutter dengan mempertimbangkan spesifikasi infrastruktur yang telah dijabarkan untuk menjalankan sistem POS ini.

UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas kehendakNya sehingga penelitian tentang perancangan sistem *Point of Sale (POS)* ini dapat diselesaikan dengan baik. Ucapan terima kasih diberikan kepada orang tua penulis yang selalu mendukung secara materil maupun non materil. Terima kasih diberikan kepada Ibu Retno Mumpuni, S.Kom. M.Sc., dan Ibu Hazna At Thooriqoh, S.Tr.Kom.,M.Kom., sebagai dosen pembimbing penulis yang senantiasa memberikan kiritikan yang memotivasi dan membangun untuk keberhasilan penelitian ini. Terima kasih juga diberikan kepada mitra apotek SF yang memberikan izin pada perusahaannya menjadi studi kasus penelitian ini. Dan ucapan terima kasih untuk Gavrila Reyna yang senantiasa memotivasi dan mendukung selama proses penulisan penelitian ini.

DAFTAR PUSTAKA

- Apriyani, Y., Kusmira, M., Iskandar, I. D., Amirulloh, I., Pertiwi, M. W., & Wibisono, T. (2022). Sistem Aplikasi Point of Sale berbasis Desktop pada Qini Mart Tasikmalaya. *Edumatic: Jurnal Pendidikan Informatika*, 6(1), 150–159. <https://doi.org/10.29408/edumatic.v6i1.5617>
- Haviluddin, H. (2016). Memahami Penggunaan UML (Unified Modelling Language). *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, 6(1), 1–15. <https://doi.org/10.30872/jim.v6i1.16>
- Khairani, R. N., Latifah, E., & Septianingrum, N. M. A. N. (2021). Evaluasi Obat Kadaluwarsa, Obat Rusak dan Stok Mati di Puskesmas Wilayah Magelang. *JURNAL FARMASI DAN ILMU KEFARMASIAN INDONESIA*, 8(1), 91–97. <https://doi.org/10.20473/jfiki.v8i12021.91-97>
- Mahfuzhi, A. R. W., Hidayah, A. K., Rifqo, M. H., & Justicia, A. S. (2025). Perancangan Sistem Informasi Kelurahan Berbasis Website Dengan Metode Object Oriented Analysis And Design (Ooad), (Studi Kasus Kelurahan Masmambang, Kabupaten Seluma). *JURNAL MEDIA INFOTAMA*, 21(2), 716–725. <https://doi.org/10.37676/jmi.v21i2.9308>
- Masula, F., & Winarno, A. (2025). Implementasi Digital Marketing dan Digitalisasi Manajemen Apotek Sugih Waras Mojokerto. *JURPIKAT*, 6(2), 1180–1192. <https://doi.org/10.37339/jurpikat.v6i2.2410>
- Narulita, S., Nugroho, A., & Abdillah, M. Z. (2024). Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS). *Bridge: Jurnal Publikasi Sistem Informasi Dan Telekomunikasi*, 2(3), 244–256. <https://doi.org/10.62951/bridge.v2i3.174>

- Nugraha, A. (2021). Perancangan Aplikasi Point Of Sales (POS) Pada Apotek Mitra Sejahtera Berbasis Web. *IKRA-ITH Informatika : Jurnal Komputer Dan Informatika*, 5(1), 74–81.
- Nugraha, S. H., & Suendri, S. (2024). Implementasi Metode Supply Chain Management pada Sistem Informasi Manajemen UMSU (Universitas Muhammadiyah Sumatera Utara) Press Bebasis Web. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 7(1), 272–280. <https://doi.org/10.32493/jtsi.v7i1.38257>
- Nurshadrina, N., & Voutama, A. (2022). *Penerapan Unified Modeling Language.(UML) Dalam Membangun Sistem Pengenalan UMKM (Studi Kasus Rafa Laundry) | INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS: Journal of Information Management*. 7(1), 21–30. <https://doi.org/https://doi.org/10.51211/imbi.v7i1.>□
- Puspaningrum, H., Anugrah, I. G., & Witra, W. P. P. (2025). PERANCANGAN PROSES INTEGRASI SISTEM INFORMASI AKADEMIK (SIKAD) DAN SISTEM PEMBELAJARAN DARING (SPADA) MENGGUNAKAN METODE OOAD (OBJECT ORIENTED ANALYSIS AND DESIGN). *JURNAL TEKNOLOGI INFORMASI DAN KOMUNIKASI*, 16(1), 120–137. <https://doi.org/10.51903/jtikp.v16i1.995>
- Putri, H. J., & Murhayati, S. (2025). Metode Pengumpulan Data Kualitatif. *Jurnal Pendidikan Tambusai*, 9(2), 13074–13086. <https://doi.org/10.31004/jptam.v9i2.27063>
- Rachmatika, P. A., Ain, R. N., Wahyudinarti, E., & Fitri, A. S. (2025). PENERAPAN METODE OBJECT ORIENTED ANALYSIS AND DESIGN PADA APLIKASI SISTEM INFORMASI PELAYANAN MASYARAKAT SURABAYA “MYSURABAYA.” *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(1). <https://doi.org/10.23960/jitet.v13i1.5829>
- Rafif, S. N., Hendriani, R., & Septyadi, R. (2025). Evaluasi Pengelolaan Obat Rusak, Kadaluarsa, dan Dead Stock Apotek X di Kota Bandung Periode Januari—Maret 2025. *FASKES : Jurnal Farmasi, Kesehatan, Dan Sains*, 3(1), 23–32. <https://doi.org/10.32665/faskes.v3i1.4474>
- Rahmawati, R., Habsy, B. A., & Nursalim, M. (2025). Jenis-Jenis Metode Pengumpulan Data (Qualitative Research). *Jurnal Pendidikan Tambusai*, 9(1), 9932–9938. <https://doi.org/10.31004/jptam.v9i1.26166>
- Setiaji, S., & Sastra, R. (2021). Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Penggajian. *Jurnal Teknik Komputer*, 7(1), 106–111. <https://doi.org/10.31294/jtk.v7i1.9773>
- Zulkarnain, Z., & Novita, R. (2023). Sistem Informasi Keuangan menggunakan Konsep Object Oriented Analysis Design (OOAD: Financial Information System Use Object Oriented Analysis Design (OOAD). *Indonesian Journal of Informatic Research and Software Engineering (IJIRSE)*, 3(2), 136–143. <https://doi.org/10.57152/ijirse.v3i2.961>