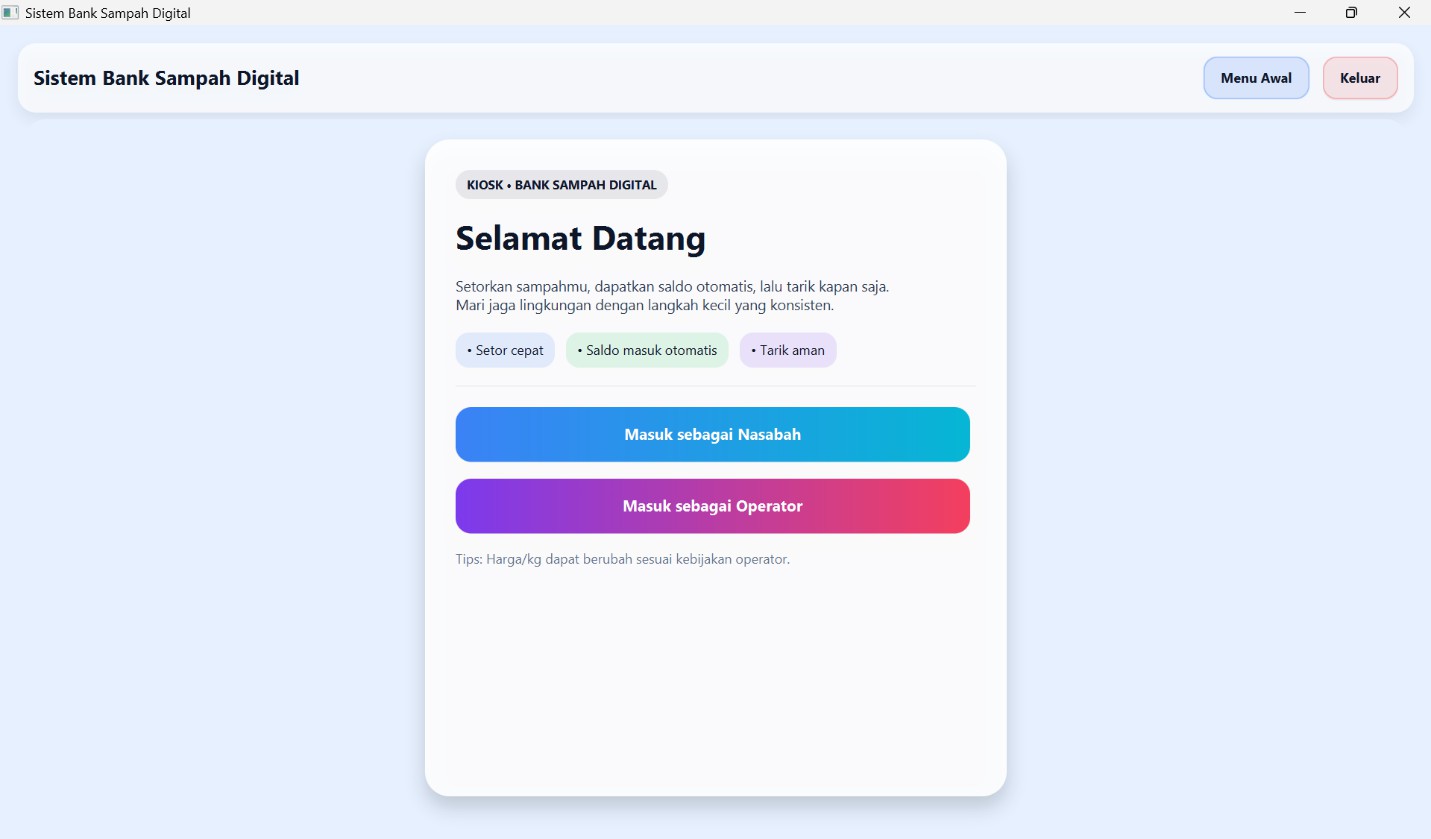
# LAPORAN PUBLIKASI

# SISTEM BANK SAMPAH DIGITAL

Disusun untuk memenuhi tugas projek mata kuliah Pemrograman Berorientasi Objek Dosen Pengampu: Ali Tarmuji, S.T., M.Cs.



Disusun oleh:

|  |  |
| --- | --- |
| Nama : | Mutiara Avivah |
| NIM : | 2400018105 |
| Kelas : | B |

Link Github:

[https://github.com/mutiaraavivah/PBO2025\_B\_105\_MUTIARA-AVIVAH\_SISTEM-BANK-](https://github.com/mutiaraavivah/PBO2025_B_105_MUTIARA-AVIVAH_SISTEM-BANK-SAMPAH-DIGITAL) [SAMPAH-DIGITAL](https://github.com/mutiaraavivah/PBO2025_B_105_MUTIARA-AVIVAH_SISTEM-BANK-SAMPAH-DIGITAL)

**PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INDUSTRI UNIVERSITAS AHMAD DAHLAN**

**YOGYAKARTA 2025/2026**

1. **Ruang Lingkup Aplikasi**

Aplikasi Sistem Bank Sampah Digital merupakan aplikasi desktop yang dikembangkan menggunakan bahasa pemrograman Java dengan framework JavaFX. Aplikasi ini bertujuan untuk menstimulasikan sistem pengelolaan bank sampah secara digital, mulai dari proses pendaftaran nasabah, transaksi setor sampah, penarikan saldo, hingga pengelolaan data oleh operator.

Ruang lingkup aplikasi dibatasi pada dua jenis pengguna utama, yaitu Nasabah dan operator. Nasabah memiliki akses untuk melakukan login, melihat saldo, menyetorkan sampah, menarik saldo, serta melihat riwayat transaksi pribadi. Sementara itu, operator bertanggung jawab dalam mengelola jenis sampah, memperbarui harga per kilogram, serta memantau ringkasan transaksi secara keseluruhan.

Aplikasi ini tidak menggunakan database eksternal, melainkan menyimpan data secara sementara menggunakan struktur data dalam memori melalui kelas DataStore. Oleh karena itu, aplikasi bersifat simulative dan difokuskan pada penerapan konsep Pemrograman Berorientasi Objek.

# Desain dan Rancangan Aplikasi

* 1. **Rancangan Konseptual**

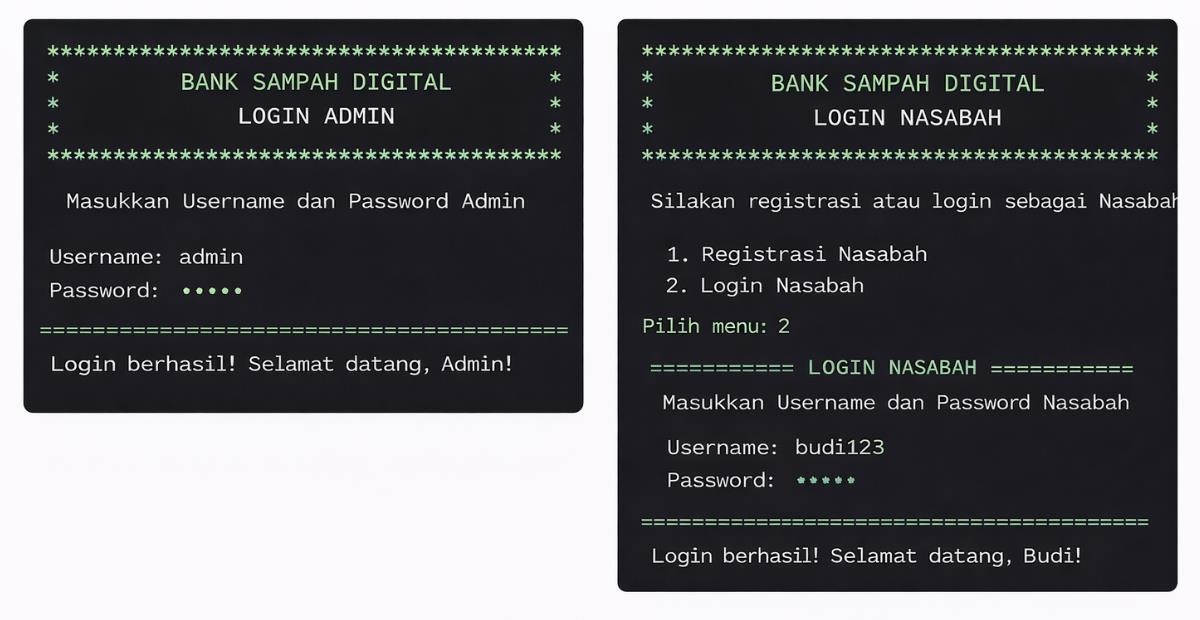
Secara konseptual, apliaksi dirancang menggunakan pendekatan Pemrograman Berorientasi Objek, dengan memisahkan antara data, logika proses, dan tampilan. Struktur aplikasi terdiri dari beberapa kelompok kelas utama, yaitu kelas model, kelas layanan (service), dan kelas antarmuka pengguna.

Kelas model digunakan untuk merepresentasikan entitas utama dalam sistem, seperti pengguna, nasabah, sampah, dan transaksi. Kelas layanan bertugas menangani logika bisnis seperti proses setor dan Tarik saldo. Sementara itu, kelas antarmuka pengguna mengatur tampilan visual aplikasi serta interaksi pengguna dengan sistem. Pendekatan ini dipilih agar kode lebih terstruktur, mudah dikembangkan, dan sesuai denga prinsip inherintance, encapsulation, dan polymorphism.

# Rancangan Antarmuka Awal (Berbasis Console)

Pada tahap awal pengembangan aplikasi sistem Bank Sampah Digital, perancangan antarmuka masih menggunakan console berbasis teks. Antarmuka console dirancang dengan tampilan teks sederhana yang dibatasi garis dan simbol agar tetap mudah dibaca oleh pengguna. Pada tampilan awal aplikasi, pengguna disajikan menu utama bertuliskan “Sistem Bank Sampah Digital” yang berisi beberapa pilihan menu, yaitu login sebagai admin, masuk ke menu nasabah, dan keluar dari aplikasi.





Pada mode Login Admin, antarmuka console menampilkan form input username dan password. Setelah data dimasukkan, sistem akan melakukan proses validasi. Jika login berhasil, sistem emnampilkan pesan konfirmasi “Login berhasil” dan mengarahkan admin ke menu pengelolaan. Sebaliknya, jika gagal, sistem menampilkan pesan kesalahan yang sesuai.

Pada mode Login Nasabah, pengguna diberikan dua pilihan, yaitu melakukan registrasi atau login. Proses registrasi memungkinkan nasabah baru memasukkan data username dan password, sedangkan proses login digunakan oleh nasabah yang telah terdaftar. Setelah login berhasil, sistem menampilkan menu nasabah yang berisi fitur setor sampah, Tarik saldo, serta melihat Riwayat transaksi.

# Rancangan Antarmuka Lanjutan (Berbasis GUI)

Seiring berjalankan proses pengerjaan projek dan meningkatnya kompleksitas aplikasi, rancangan antarmuka berbasis console dinilai memiliki keterbatasan dari sisi kenyamanan pengguna (user experience) dan visualisasi sistem. Oleh karena itu, pada tahap selanjutnya dilakukan perombakan desain antarmuka dengan mengimplementasikan Graphical User Interface (GUI) menggunakan JavaFX.

Rancangan antarmuka GUI ini dikembangkan sebagai versi lanjutan dari desain console sebelumnya, dengan tetap mempertahankan seluruh fungsi utama sistem. Pada tahap pengembangan ini, peran admin pada versi awal disesuaikan dan dikembangkan menjadi operator, yang bertugas mengelola data sistem seperti jenis sampah, harga per kilogram, serta memantau ringkasan transaksi. Selain itu, sistem tetap menyediakan fitur login dan registrasi nasabah, proses setor sampah, penarikan saldo, serta akses informasi transaksi.

Perubahan utama pada pengembangan ini terletak pada cara interaksi pengguna, yang sebelumnya berbasis teks melalui console menjadi berbasis komponen visual. Antarmuka GUI memanfaatkan elemen-elemen seperti tombol, form input, tabel, panel, dan navigasi layar, sehingga alur penggunaan sistem menjadi lebih intuitif, mudah dipahami, dan nyaman bagi pengguna.

# Diagram UML Aplikasi

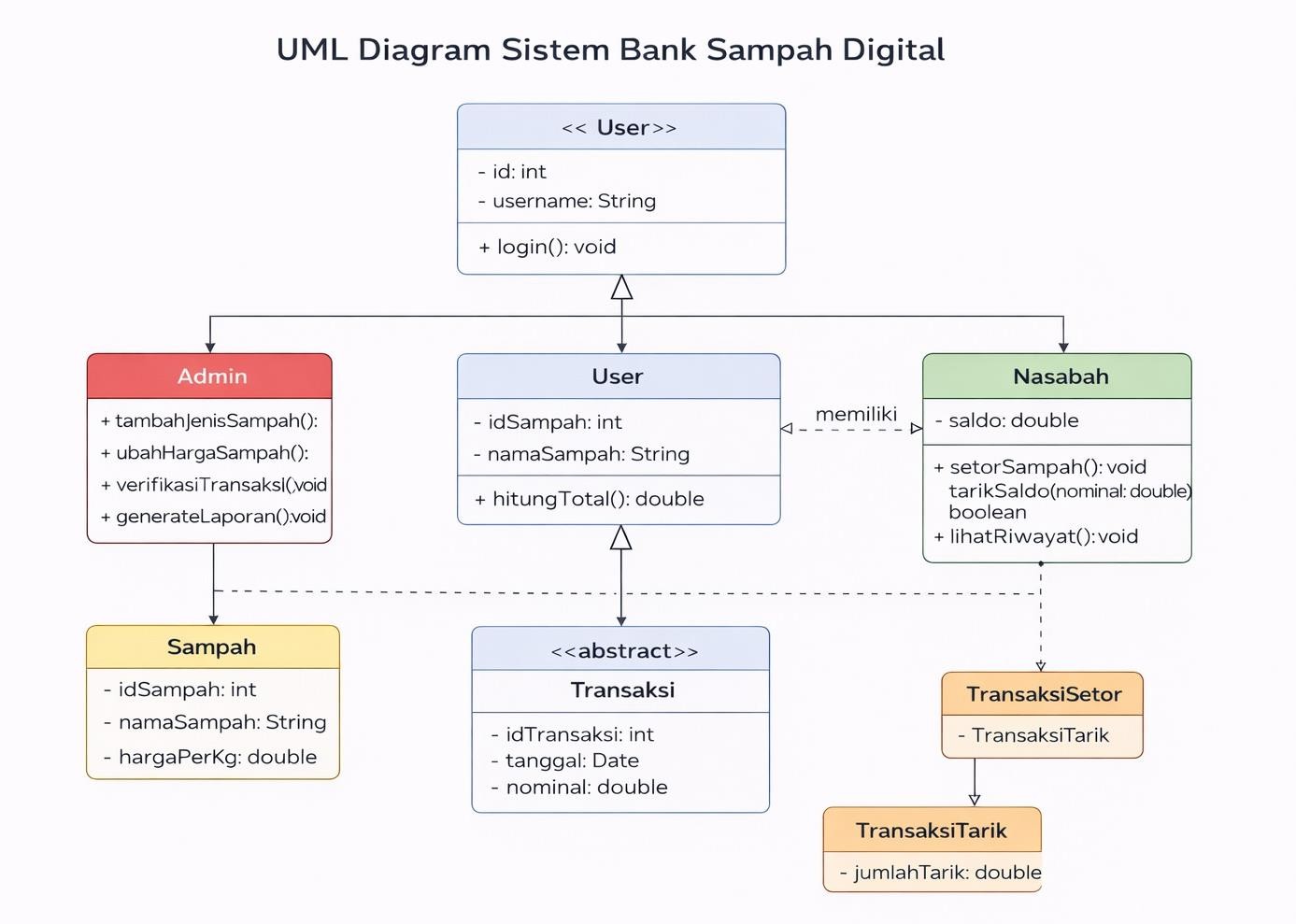
****

Diagram UML Sistem Bank Sampah Digital tersebut merepresentasikan struktur kelas dan hubungan antarobjek yang membentuk alur kerja sistem secara konseptual. Secara umum, sistem ini dirancang untuk mengelola aktivitas pengelolaan sampah berbasis digital yang melibatkan peran pengguna, pengelola, transaksi, serta data sampah, dengan pendekatan berorientasi objek.

Pada lapisan paling atas terdapat kelas <<User>> yang berfungsi sebagai superclass atau kelas induk. Kelas ini merepresentasikan entitas pengguna secara umum dalam sistem dan memiliki atribut dasar berupa id dan username. Selain itu, kelas ini menyediakan operasi login() yang menjadi fungsi dasar autentikasi bagi seluruh turunan kelasnya. Dengan adanya superclass ini, prinsip pewarisan (inheritance) diterapkan untuk menyederhanakan struktur dan menghindari duplikasi atribut maupun metode yang bersifat umum.

Kelas Admin dan Nasabah merupakan turunan dari kelas <<User>>. Kelas Admin memiliki tanggung jawab utama dalam pengelolaan sistem, khususnya yang berkaitan dengan data dan validasi. Metode tambahJenisSampah() dan ubahHargaSampah() menunjukkan peran admin dalam mengelola data sampah, sedangkan verifikasiTransaksi() digunakan untuk memastikan keabsahan transaksi yang dilakukan oleh nasabah. Selain itu, metode generateLaporan() berfungsi untuk menghasilkan laporan operasional sistem, yang umumnya digunakan untuk evaluasi dan pengambilan keputusan.

Kelas Nasabah merepresentasikan pengguna yang melakukan aktivitas utama dalam bank sampah digital. Nasabah memiliki atribut saldo yang mencerminkan nilai tabungan hasil setoran sampah. Operasi setorSampah() digunakan ketika nasabah menyerahkan sampah ke sistem, sedangkan tarikSaldo(nominal) memungkinkan nasabah menarik saldo yang dimiliki dengan pengembalian nilai boolean sebagai indikator keberhasilan transaksi. Metode lihatRiwayat() disediakan untuk menampilkan riwayat transaksi yang pernah dilakukan oleh nasabah, sehingga mendukung transparansi dan akuntabilitas.

Kelas Sampah berfungsi sebagai entitas data yang menyimpan informasi terkait jenis sampah. Atribut idSampah, namaSampah, dan hargaPerKg digunakan untuk mengidentifikasi sampah dan menentukan nilai ekonominya. Kelas ini berelasi dengan admin, yang berarti pengelolaan data sampah sepenuhnya berada di bawah kewenangan admin. Relasi ini menunjukkan bahwa perubahan data sampah tidak dapat dilakukan secara langsung oleh nasabah.

Kelas Transaksi merupakan kelas abstrak yang menjadi dasar bagi seluruh jenis transaksi dalam sistem. Atribut idTransaksi, tanggal, dan nominal merepresentasikan informasi umum yang dimiliki oleh setiap transaksi. Sebagai kelas abstrak, Transaksi tidak dapat diinstansiasi secara langsung, melainkan diturunkan ke kelas-kelas konkret yang merepresentasikan jenis transaksi tertentu. Pendekatan ini mencerminkan penerapan prinsip abstraksi dalam pemodelan sistem.

Turunan dari kelas Transaksi adalah TransaksiSetor dan TransaksiTarik. TransaksiSetor merepresentasikan transaksi penyetoran sampah oleh nasabah yang akan menambah saldo. Sementara itu, TransaksiTarik merepresentasikan transaksi penarikan saldo, dengan atribut tambahan jumlahTarik untuk mencatat nominal dana yang ditarik. Hubungan antara nasabah dan transaksi digambarkan melalui asosiasi, yang menunjukkan bahwa seorang nasabah dapat memiliki banyak transaksi, baik setor maupun tarik.

# Hasil Akhir Aplikasi

* 1. **Perbedaan Progres Awal dan Hasil Akhir**

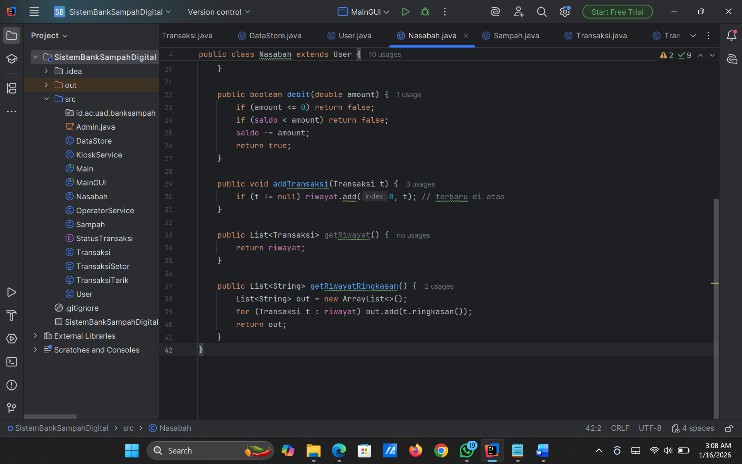
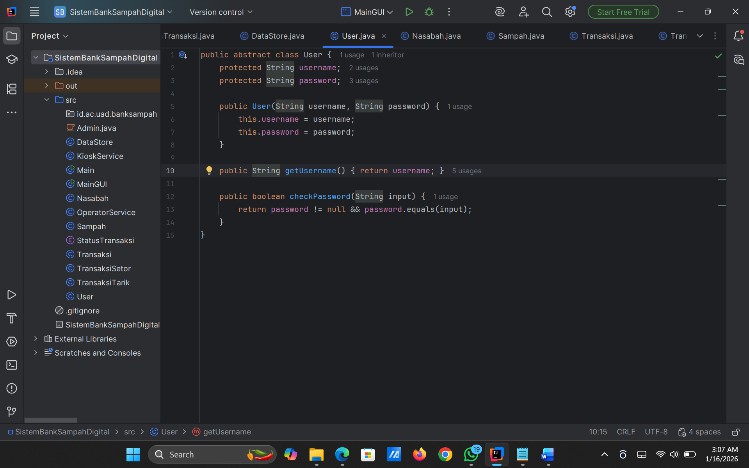
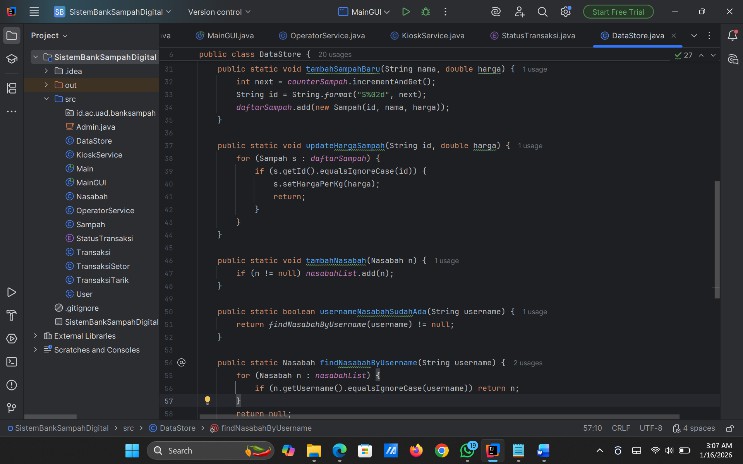
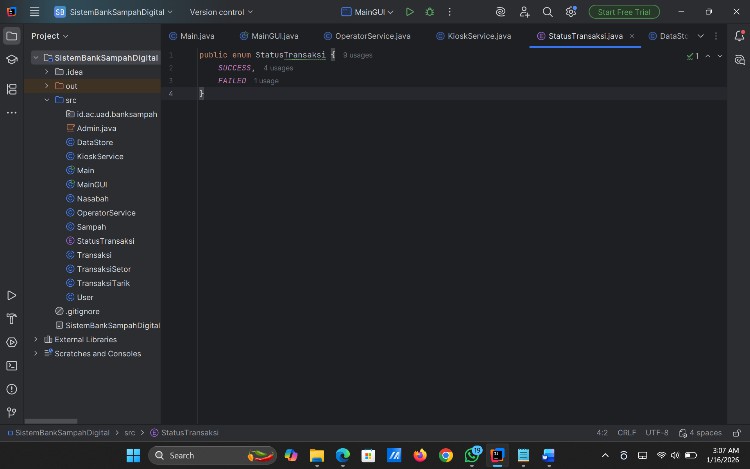
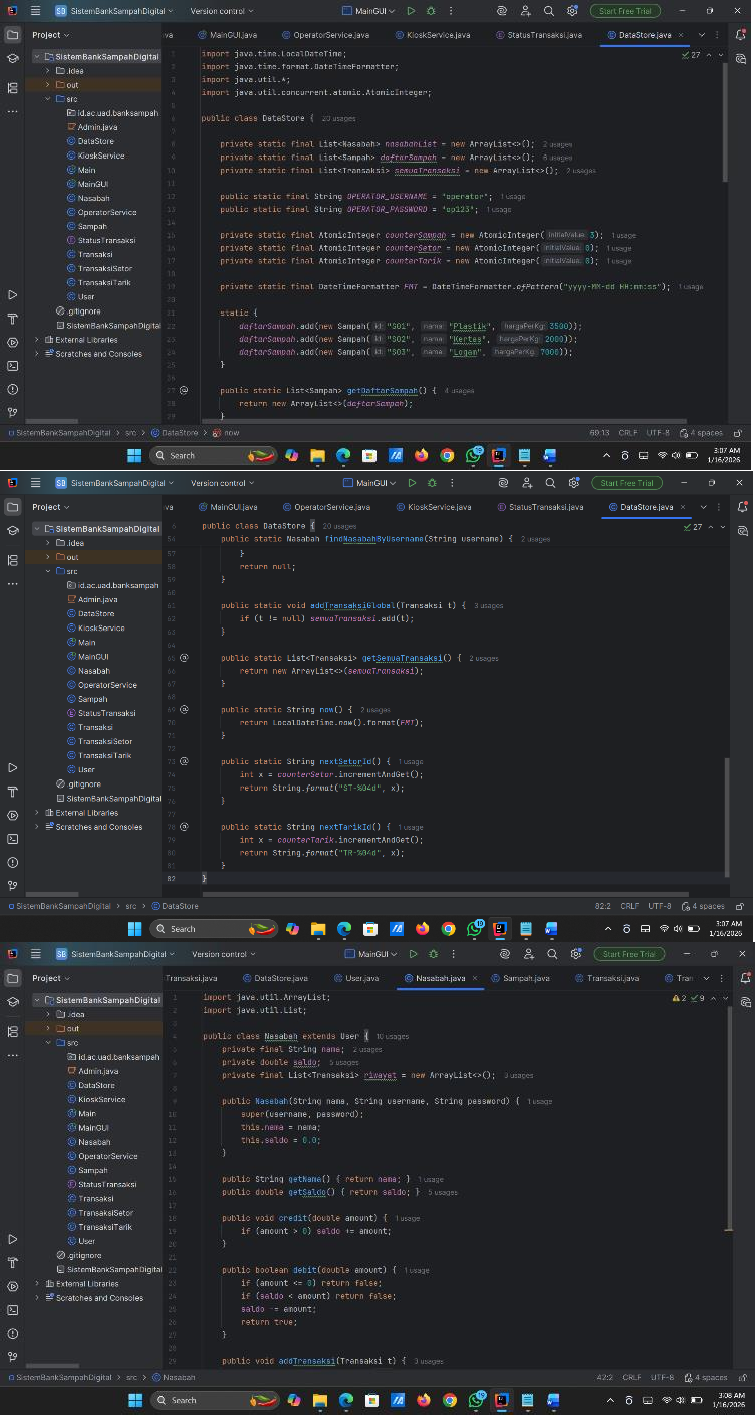
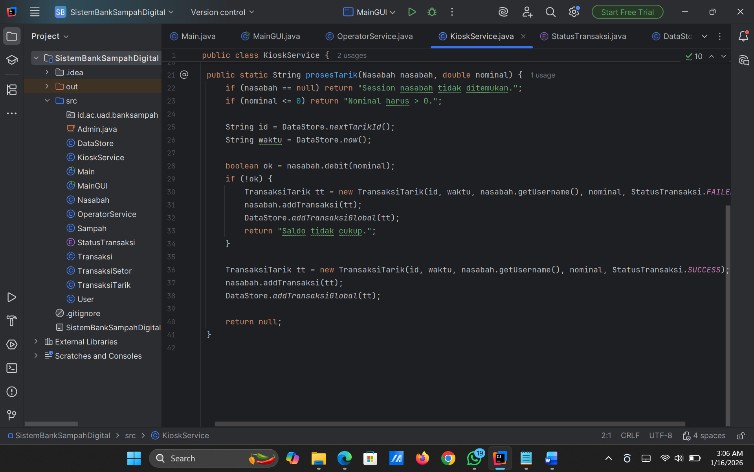
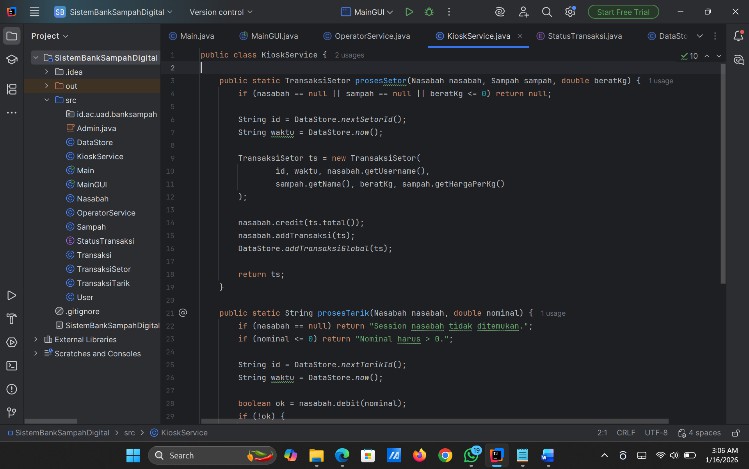
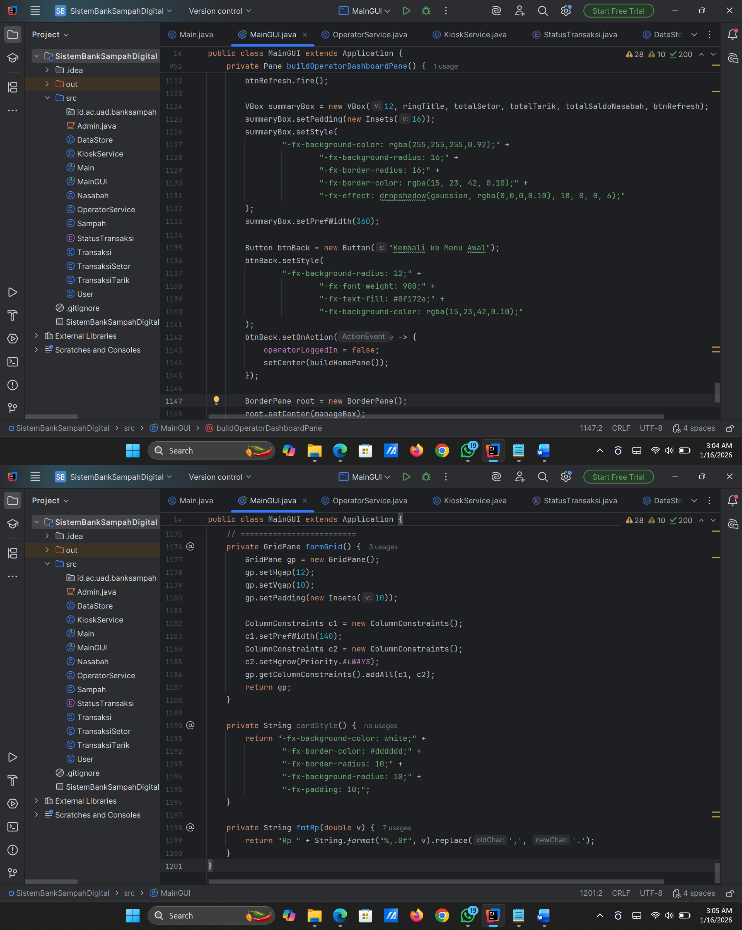
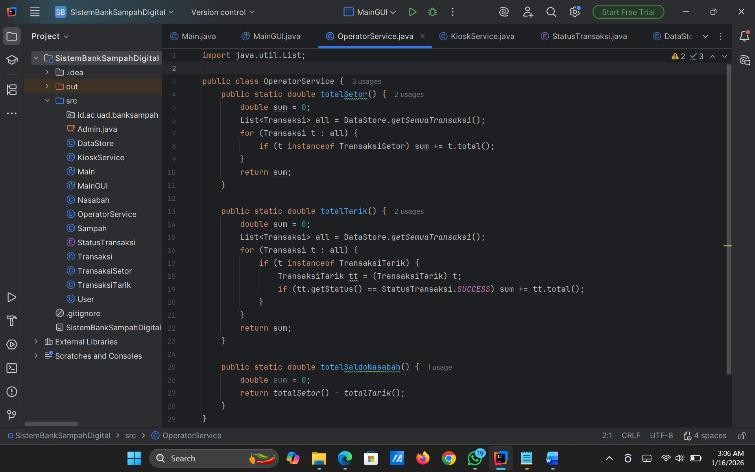
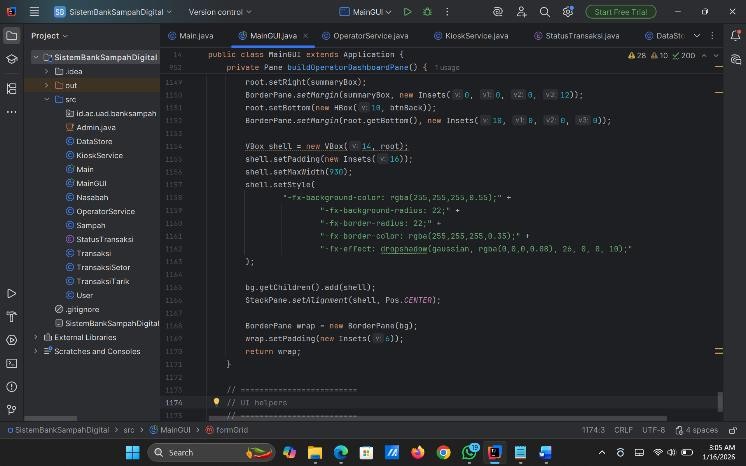
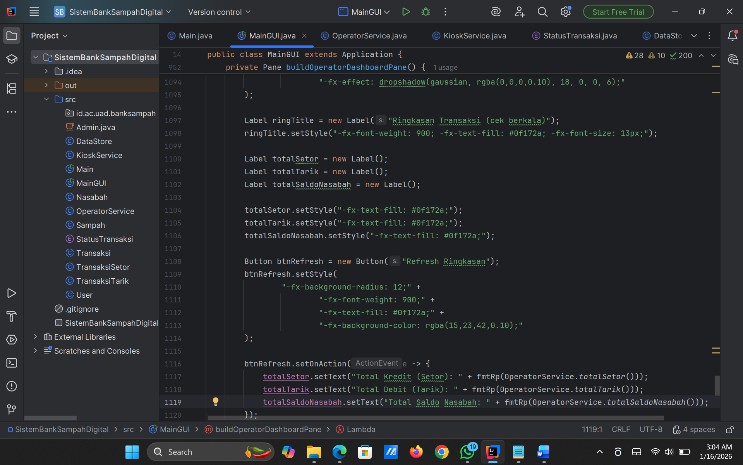
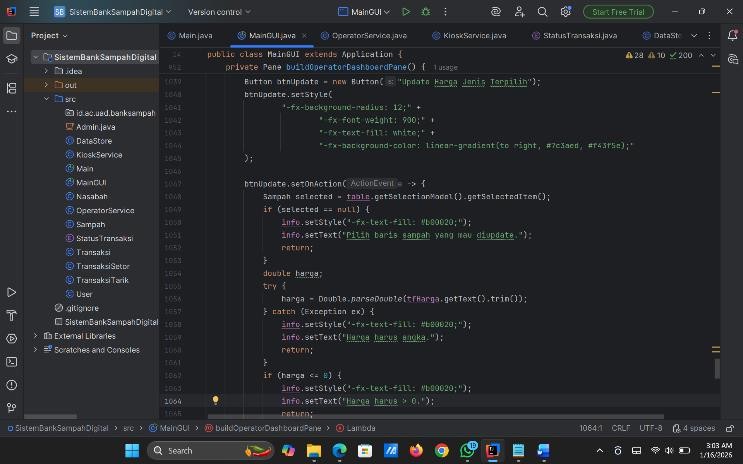
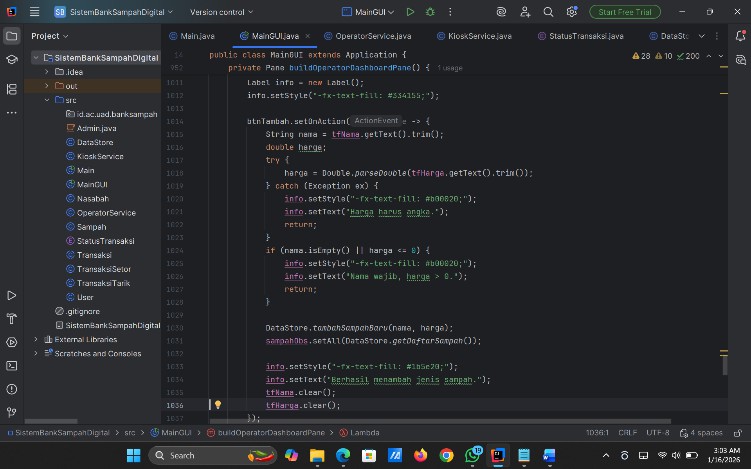
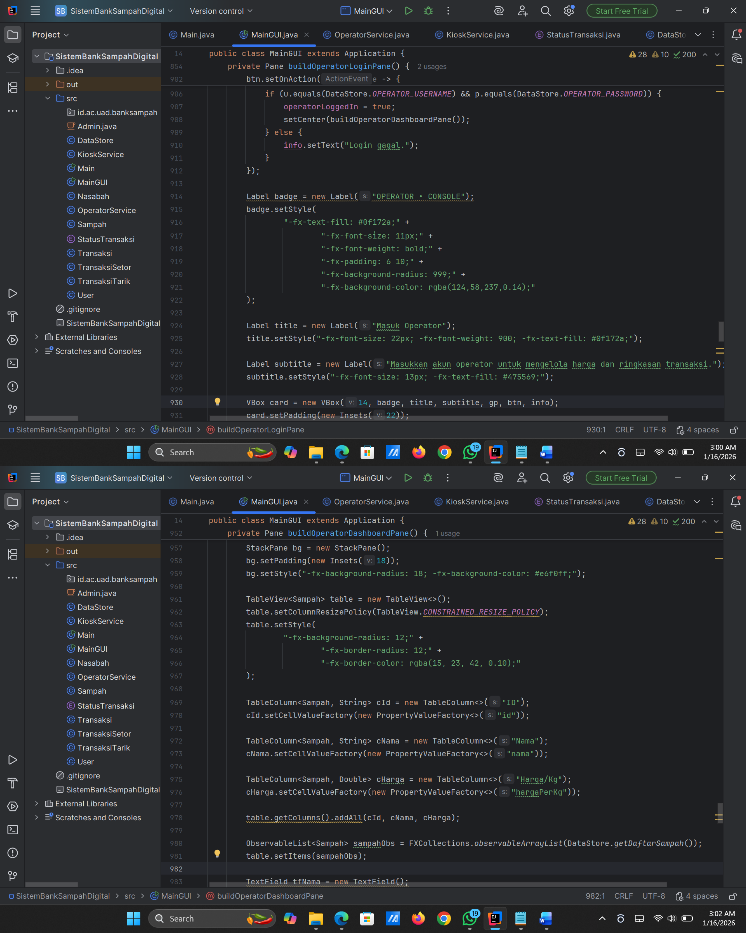
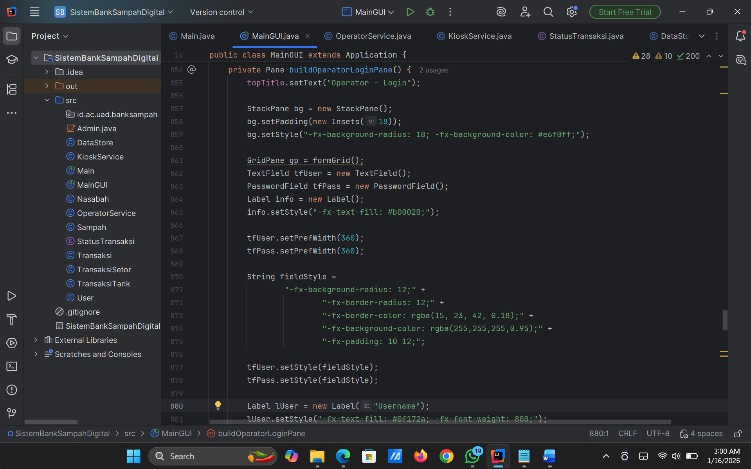
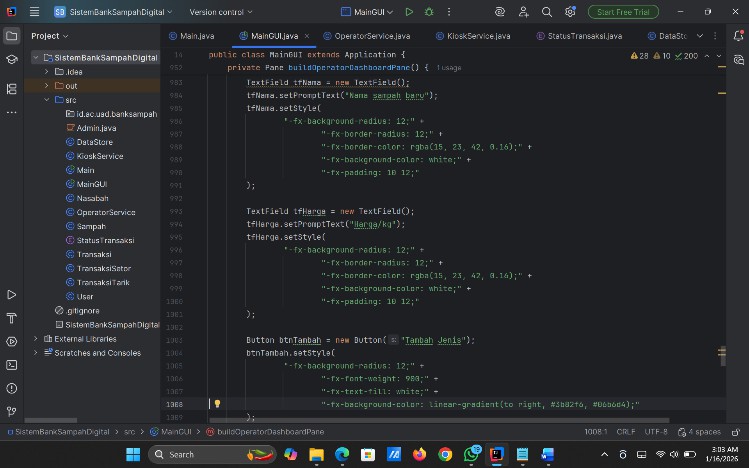
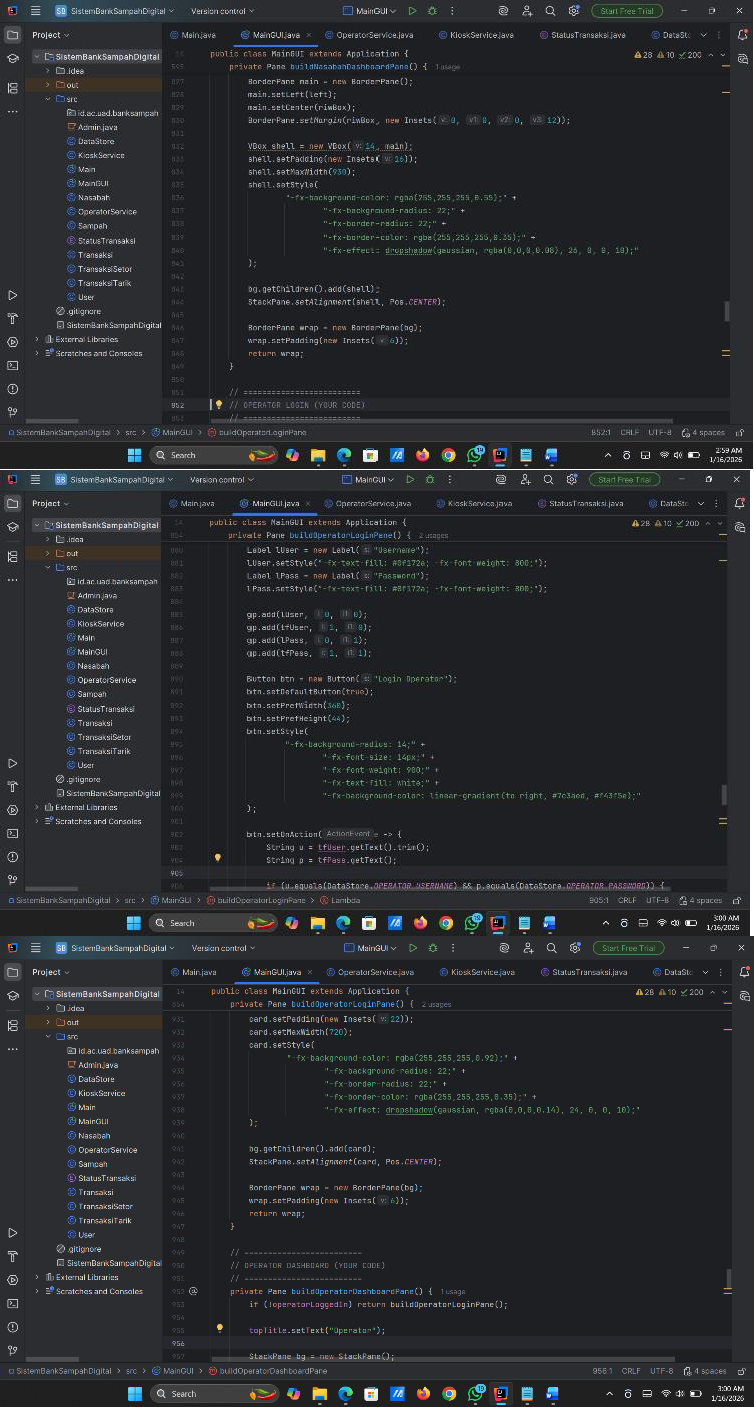
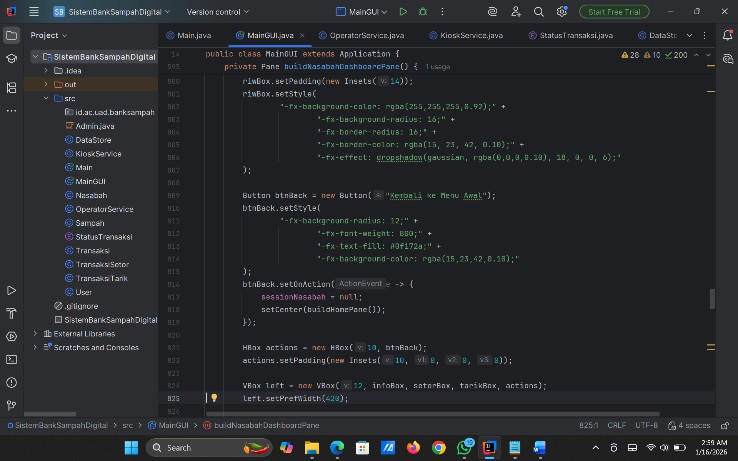
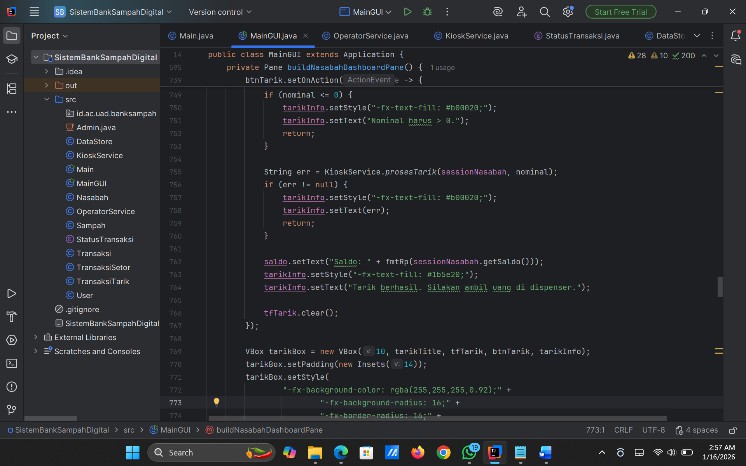
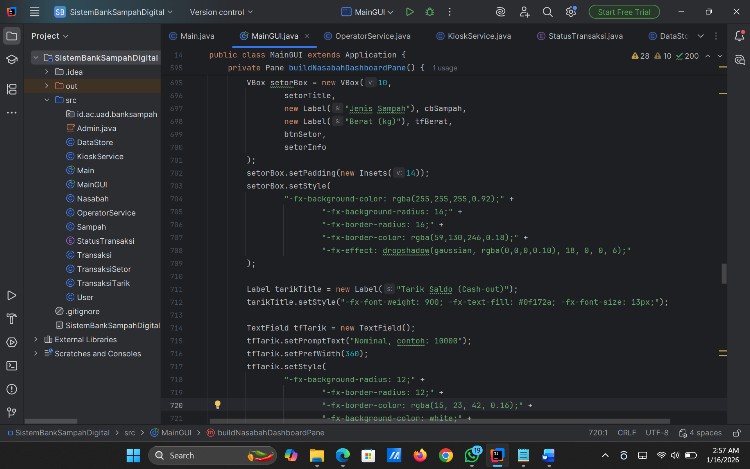
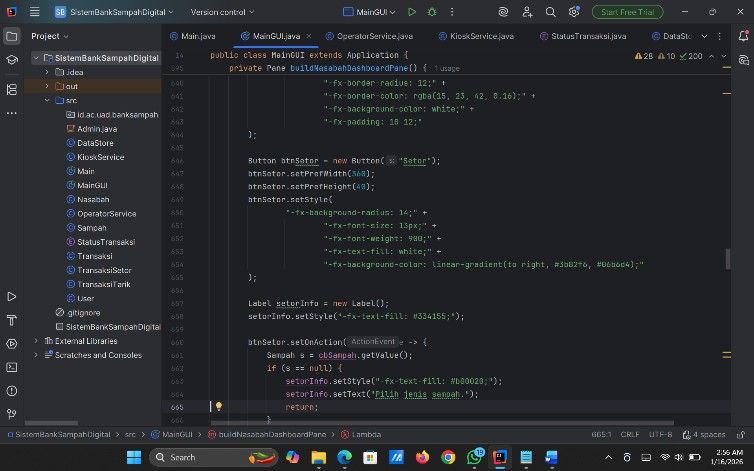
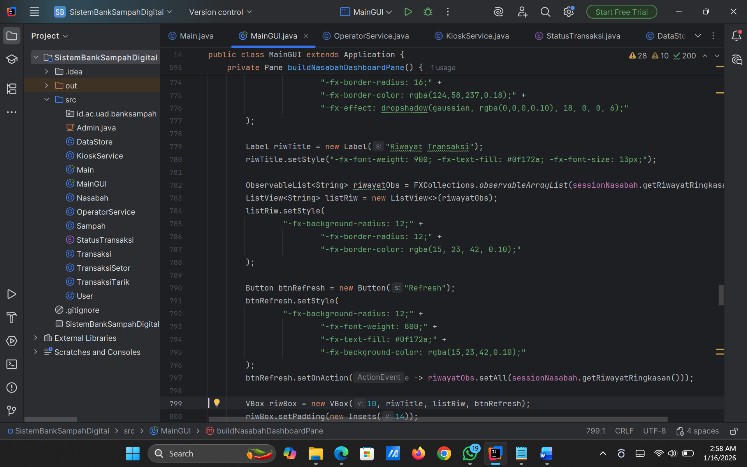
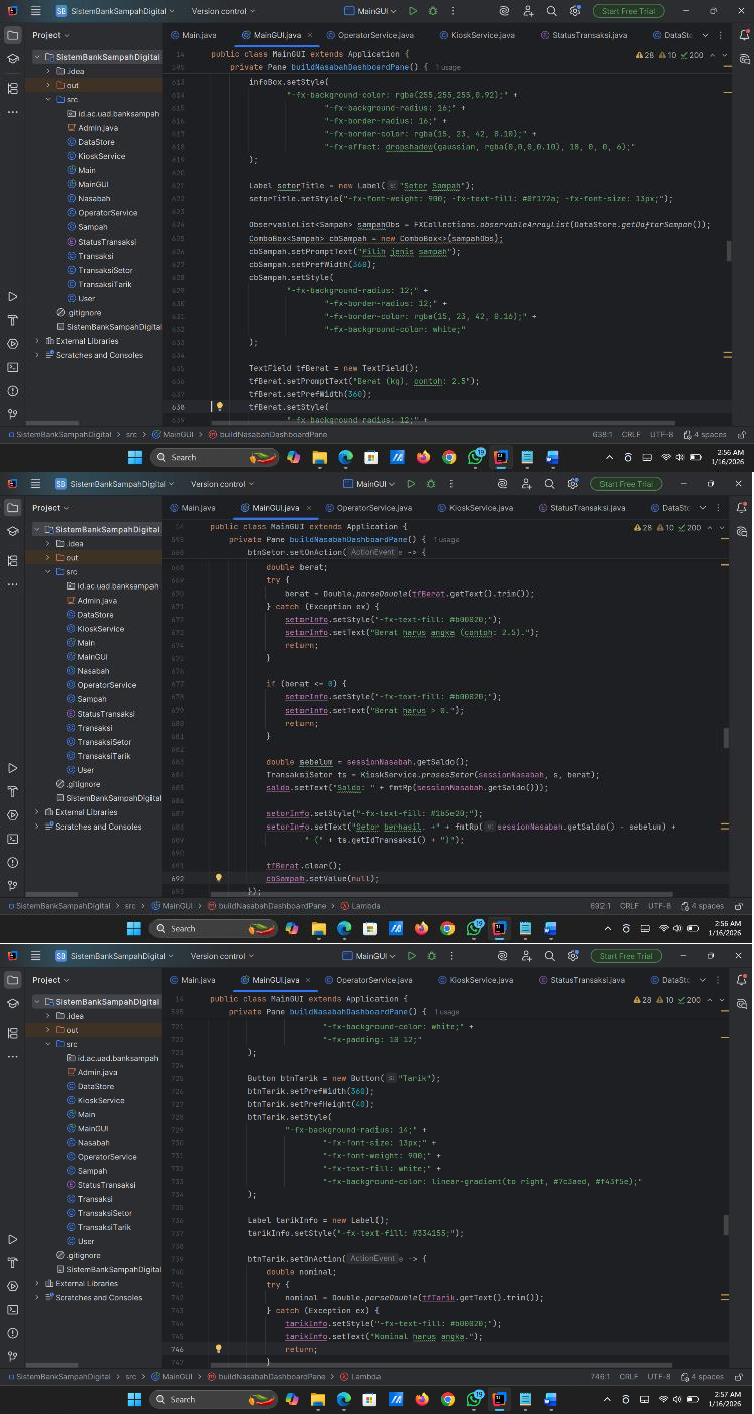
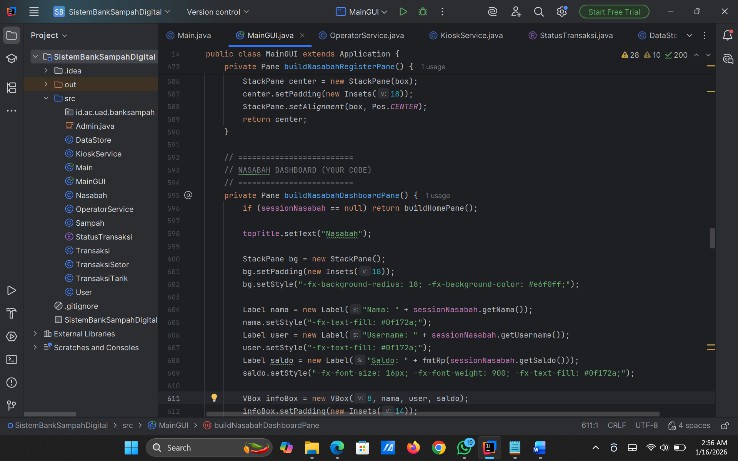
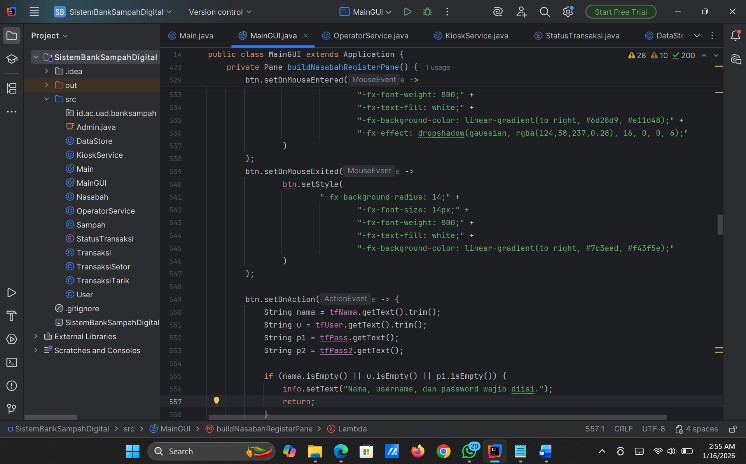
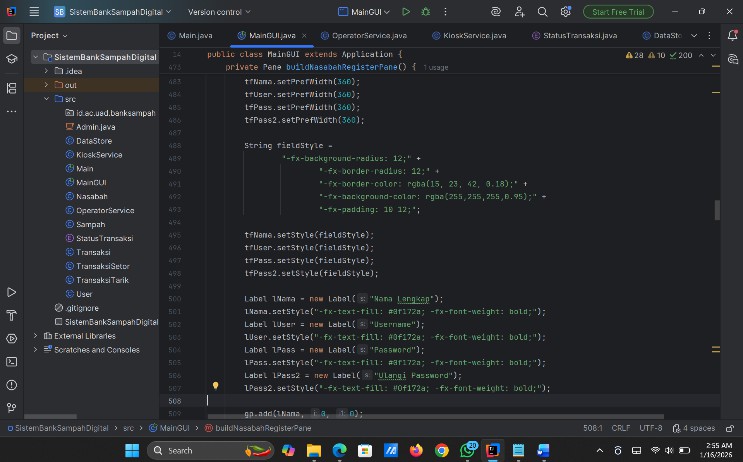
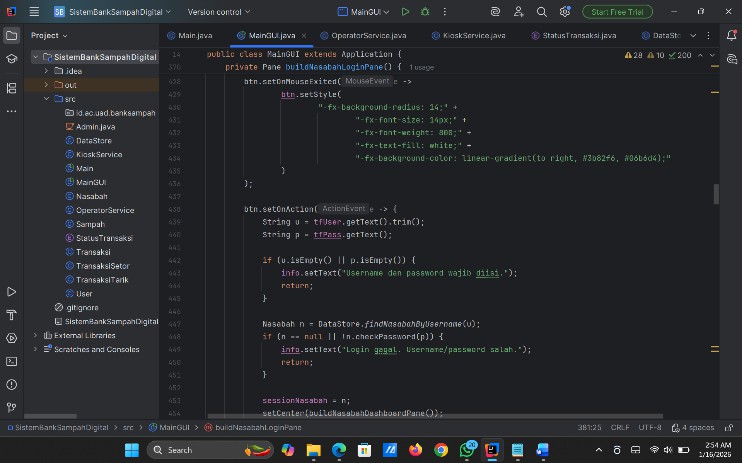
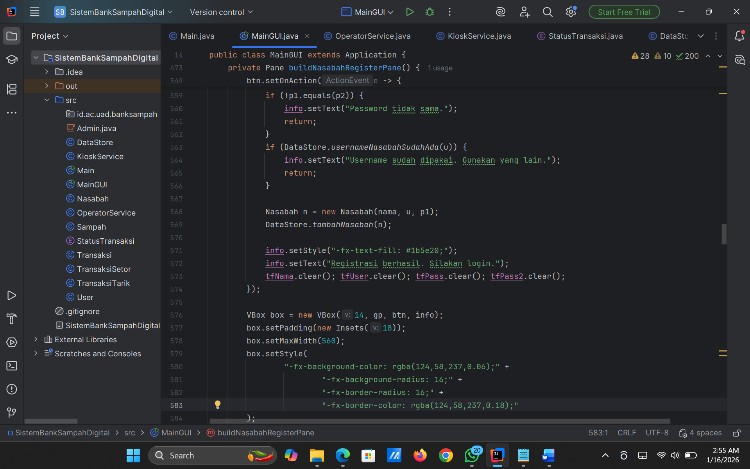
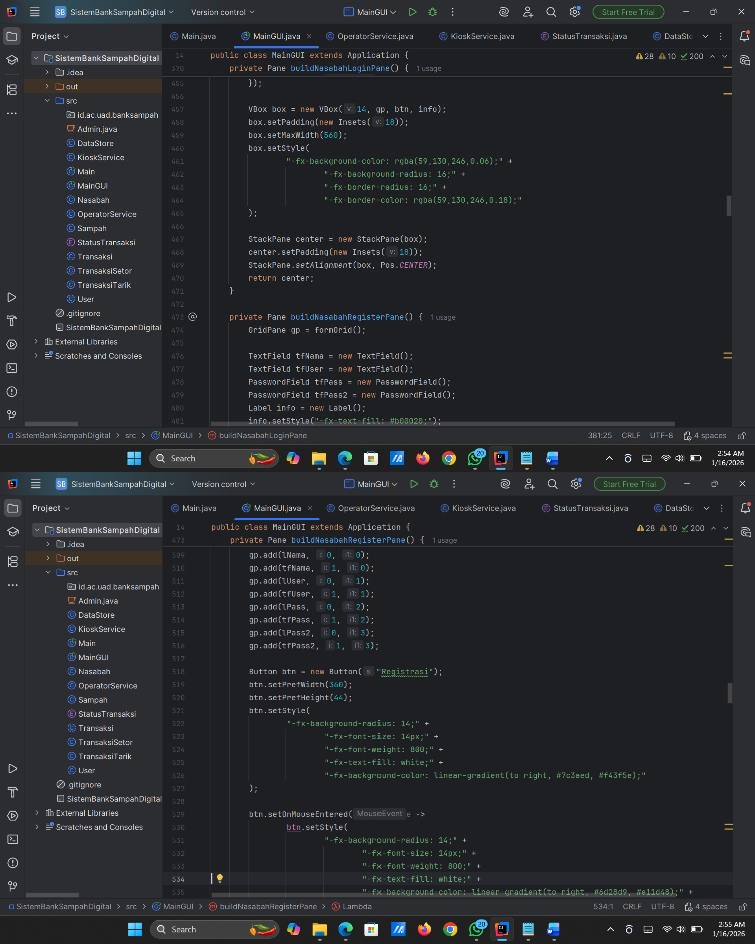
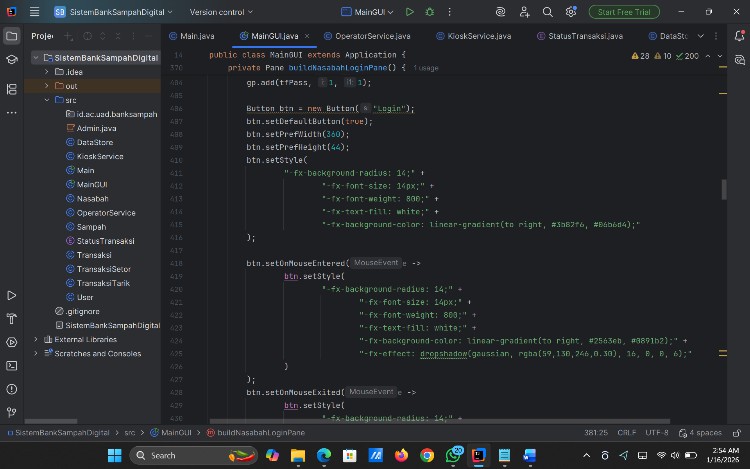
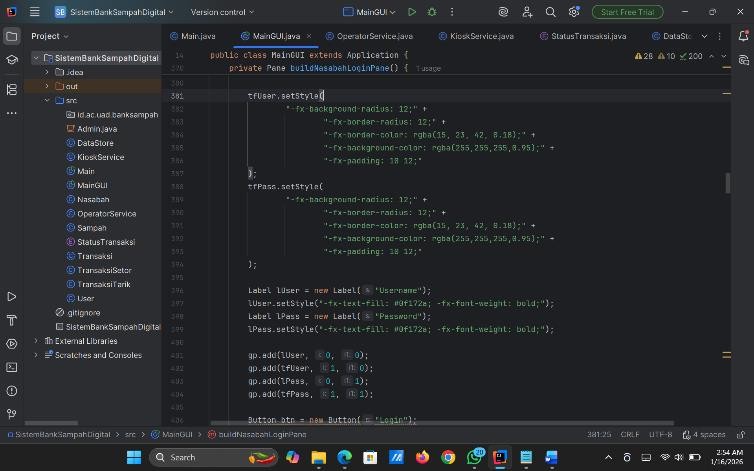
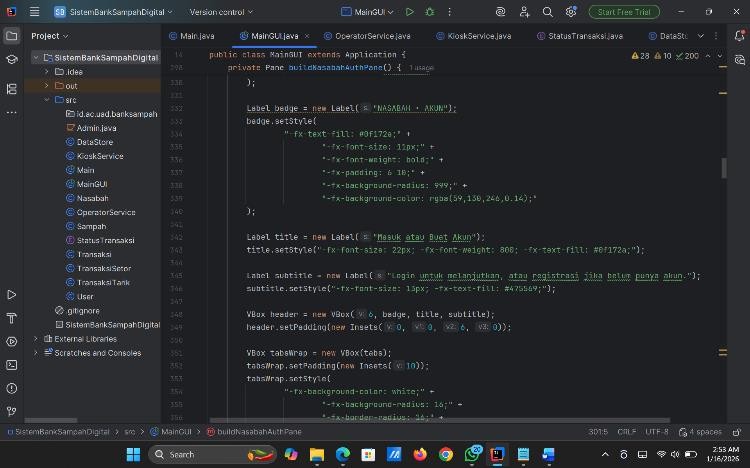
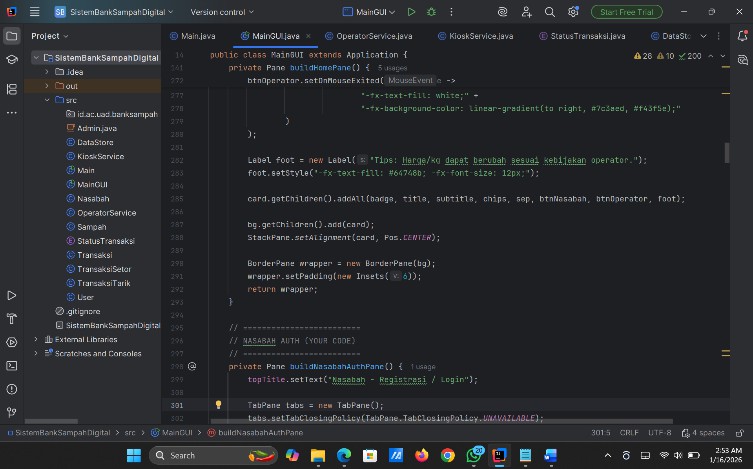
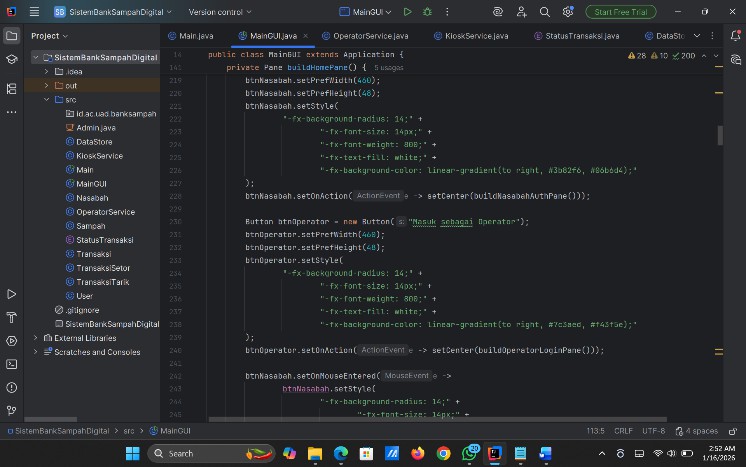
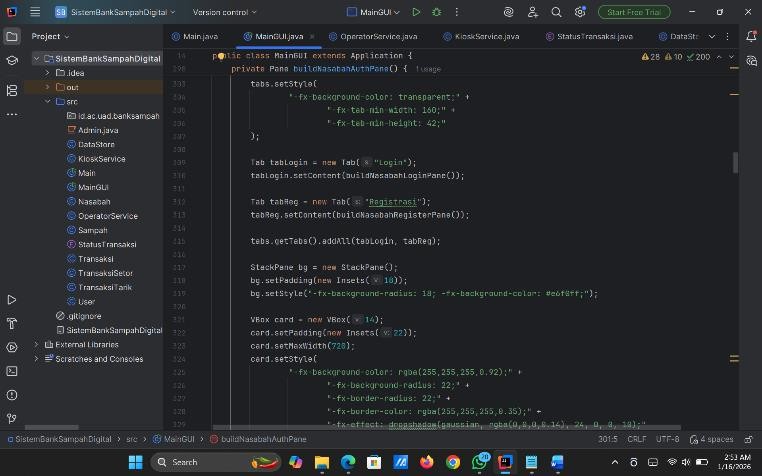
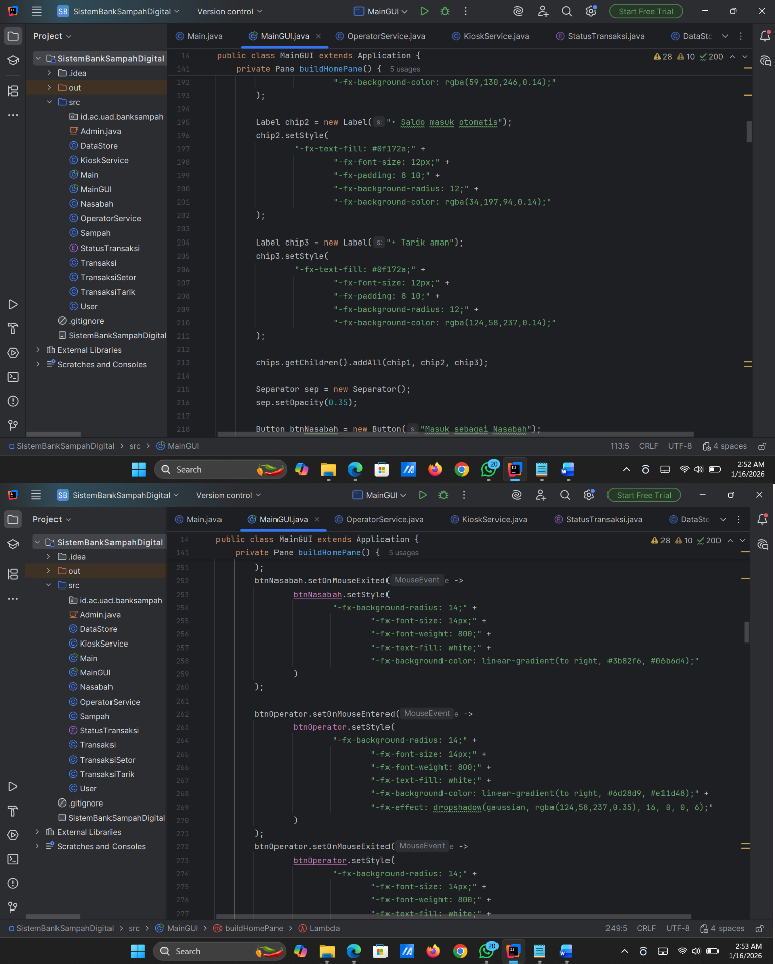
Pada tahap progress awal, aplikasi masih berada pada tahap dasar. Tampilan antarmuka masih sederhana, belum konsisten antar halaman, dan belum sepenuhnya mencerminkan konsep modern. Selain itu, struktur kode masih berkembang dan beberapa fitur belum terintegrasi secara optimal.

Pada hasil akhir aplikasi, terjadi perubahan dan pengembangan yang signifikan. Hampir seluruh tampilan antarmuka dirombak ulang menggunakan konsep desain baru berbasis JavaFX. Top bar global ditambahkan agar navigasi lebih konsisten, layout aplikasi dibuat full screen sehingga dapat menyesuaikan ukuran layar pengguna tanpa mengubah proporsi komponen utama.

GUI pada aplikasi ini dirancang untuk mendukung dua peran utama, yaitu Nasabah dan Operator, yang masing-masing memiliki tampilan dan fungsi berbeda sesuai dengan kebutuhan sistem. Seluruh proses utama seperti registrasi, login, setor sampah, tarik saldo, pengelolaan harga sampah, serta pemantauan transaksi telah terintegrasi secara visual dalam antarmuka grafis.

# Tampilan Kode pada Intellij IDEA

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |



|  |  |
| --- | --- |
|  |  |
|  |  |

# Penjelasan:

* + 1. **Main.java**

Main.java berfungsi sebagai entry point aplikasi. Di dalam method main, program tidak langsung membangun tampilan, melainkan memanggil MainGUI.main(args). Pola ini dibuat agar proses eksekusi aplikasi terpusat di kelas MainGUI sebagai aplikasi JavaFX, sementara Main hanya menjadi penghubung awal ketika program dijalankan.

# MainGUI.java

MainGUI.java merupakan pusat aplikasi karena kelas ini extends pplication dan memuat seluruh mekanisme pembentukan antarmuka serta navigasi antar halaman. Di dalamnya terdapat pengaturan Stage, pembuatan root layout BorderPane, pemasangan top bar global, serta penggantian konten halaman menggunakan setCenter(Node) sehingga aplikasi cukup memakai satu window dengan isi yang berubah-ubah.

MainGUI juga menyimpan konsep sesi melalui sessionNasabah untuk menandai nasabah yang sedang login dan operatorLoggedIn untuk status operator, sehingga setelah login berhasil, data pengguna dapat ditampilkan dan dipakai untuk menjalankan transaksi tanpa harus mencari ulang dari awal. Selain logika navigasi, file ini juga memuat styling (CSS) dan pembuatan komponen penting seperti halaman home, login/registrasi nasabah, dashboard nasabah, login operator, serta dashboard operator.

# DataStore.java

DataStore berperan sebagai penyimpanan data sementara (in-memory database) selama program berjalan. Kelas ini menyimpan daftar nasabah (nasabahList), daftar jenis sampah (daftarSampah), serta seluruh transaksi global (semuaTransaksi) agar operator dapat menghitung ringkasan transaksi. DataStore juga menyediakan data awal jenis sampah pada blok static, serta menyediakan generator ID otomatis menggunakan AtomicInteger untuk sampah baru (S04, S05, dst), transaksi setor (ST-0001, dst), dan transaksi tarik (TR-0001, dst).

Selain itu, DataStore menyediakan fungsi-fungsi penting seperti menambah nasabah, mencari nasabah berdasarkan username, mengecek duplikasi username, menambah transaksi ke daftar global, mengambil daftar transaksi, serta menghasilkan timestamp transaksi dengan format waktu yang konsisten. Desain ini membuat seluruh data aplikasi terpusat dan mudah diakses oleh GUI maupun service tanpa perlu database eksternal.

# KioskService.java

KioskService berfungsi sebagai lapisan proses bisnis (business logic) untuk transaksi nasabah, yaitu setor sampah dan tarik saldo. Pada prosesSetor, service memvalidasi input dasar (nasabah, sampah, dan berat harus valid), kemudian membuat ID transaksi dan waktu dari DataStore, membentuk objek TransaksiSetor, menghitung total otomatis (berat × harga/kg), lalu menambah saldo nasabah melalui credit, menyimpan transaksi ke riwayat nasabah, dan juga menyimpan transaksi ke transaksi global agar operator dapat melihat rekap.

Pada prosesTarik, service mengecek nominal dan ketersediaan saldo; jika saldo tidak cukup maka transaksi tarik tetap dibuat dengan status FAILED agar riwayat mencatat percobaan transaksi, sedangkan jika saldo cukup maka transaksi dibuat dengan status SUCCESS dan saldo nasabah dikurangi. Metode tarik mengembalikan null ketika sukses dan string pesan ketika gagal, sehingga GUI cukup menampilkan hasil tanpa menulis aturan bisnis secara langsung di event handler.

# OperatorService.java

OperatorService bertugas melakukan perhitungan ringkasan transaksi berdasarkan daftar transaksi global yang disimpan pada DataStore. Metode totalSetor() menjumlahkan semua transaksi bertipe TransaksiSetor, sedangkan totalTarik() hanya menjumlahkan transaksi TransaksiTarik yang berstatus SUCCESS agar transaksi gagal tidak mempengaruhi total debit.

Untuk totalSaldoNasabah(), perhitungan dilakukan dengan pendekatan akumulatif sederhana yaitu total setor dikurangi total tarik sukses, karena pada versi ini DataStore tidak mengekspose daftar nasabah untuk penjumlahan saldo langsung. Dengan demikian, operator dapat memantau “arus masuk” dari setor dan “arus keluar” dari tarik secara ringkas tanpa mengakses data nasabah satu per satu.

# StatusTransaksi.java

StatusTransaksi adalah enum yang mendefinisikan status transaksi menjadi dua kondisi tetap, yaitu SUCCESS dan FAILED. Penggunaan enum mencegah kesalahan penulisan status dalam bentuk string bebas, dan membuat kondisi logika menjadi lebih aman saat mengecek hasil transaksi, misalnya ketika operator menghitung total tarik yang hanya memasukkan transaksi sukses. Dengan status yang baku, program juga lebih mudah dipelihara karena penambahan kondisi baru dapat dilakukan dengan terkontrol.

# User.java

User merupakan kelas abstrak untuk menyimpan atribut dasar akun, yaitu username dan password. Kelas ini menyediakan method checkPassword untuk memvalidasi password input, sehingga turunan kelas tidak perlu mengulang logika pengecekan password. Dengan menjadikan User sebagai abstract class, struktur program menunjukkan konsep pewarisan (inheritance) dan abstraksi: semua jenis pengguna memiliki identitas login, namun detail perilaku dan atribut tambahannya ditentukan oleh kelas turunan seperti Nasabah.

# Nasabah.java

Nasabah adalah turunan dari User yang merepresentasikan pengguna nasabah pada sistem bank sampah. Kelas ini menambahkan atribut nama, saldo, dan riwayat transaksi. Pengelolaan saldo dilakukan melalui method credit (menambah saldo) dan debit (mengurangi saldo dengan syarat saldo cukup), sehingga aturan saldo berada di satu tempat dan tidak tersebar di GUI.

Riwayat transaksi disimpan dalam list dan transaksi terbaru dimasukkan di posisi awal (add(0, t)) agar urutan riwayat selalu dari yang paling baru. Selain itu, method getRiwayatRingkasan() mengubah daftar transaksi menjadi daftar string ringkasan yang siap ditampilkan di GUI.

# Sampah.java

Sampah memodelkan objek jenis sampah yang dapat disetorkan, dengan atribut id, nama, dan hargaPerKg. Harga dibuat dapat diubah agar operator bisa melakukan update harga berdasarkan kebijakan. Method toString() dioverride agar ketika objek Sampah tampil di ComboBox, pengguna melihat format yang informatif (nama + harga/kg), bukan sekadar alamat objek.

Dengan demikian, file ini berperan penting sebagai representasi data yang dipakai baik pada proses setor nasabah maupun pengelolaan jenis sampah oleh operator.

# Transaksi.java

Transaksi adalah kelas abstrak yang menjadi dasar semua transaksi dalam sistem. Atribut umum seperti idTransaksi, waktu, usernameNasabah, dan status disediakan agar setiap transaksi memiliki identitas yang konsisten dan dapat ditelusuri. Kelas ini memaksa turunan untuk mengimplementasikan method total() untuk nilai rupiah transaksi dan ringkasan() untuk format teks ringkas transaksi. Dengan desain ini, program menerapkan konsep polimorfisme: GUI atau service dapat memproses transaksi secara umum tanpa harus mengetahui detail tipe transaksi, karena setiap transaksi pasti mampu menghitung total dan menghasilkan ringkasan sendiri.

# TransaksiSetor.java

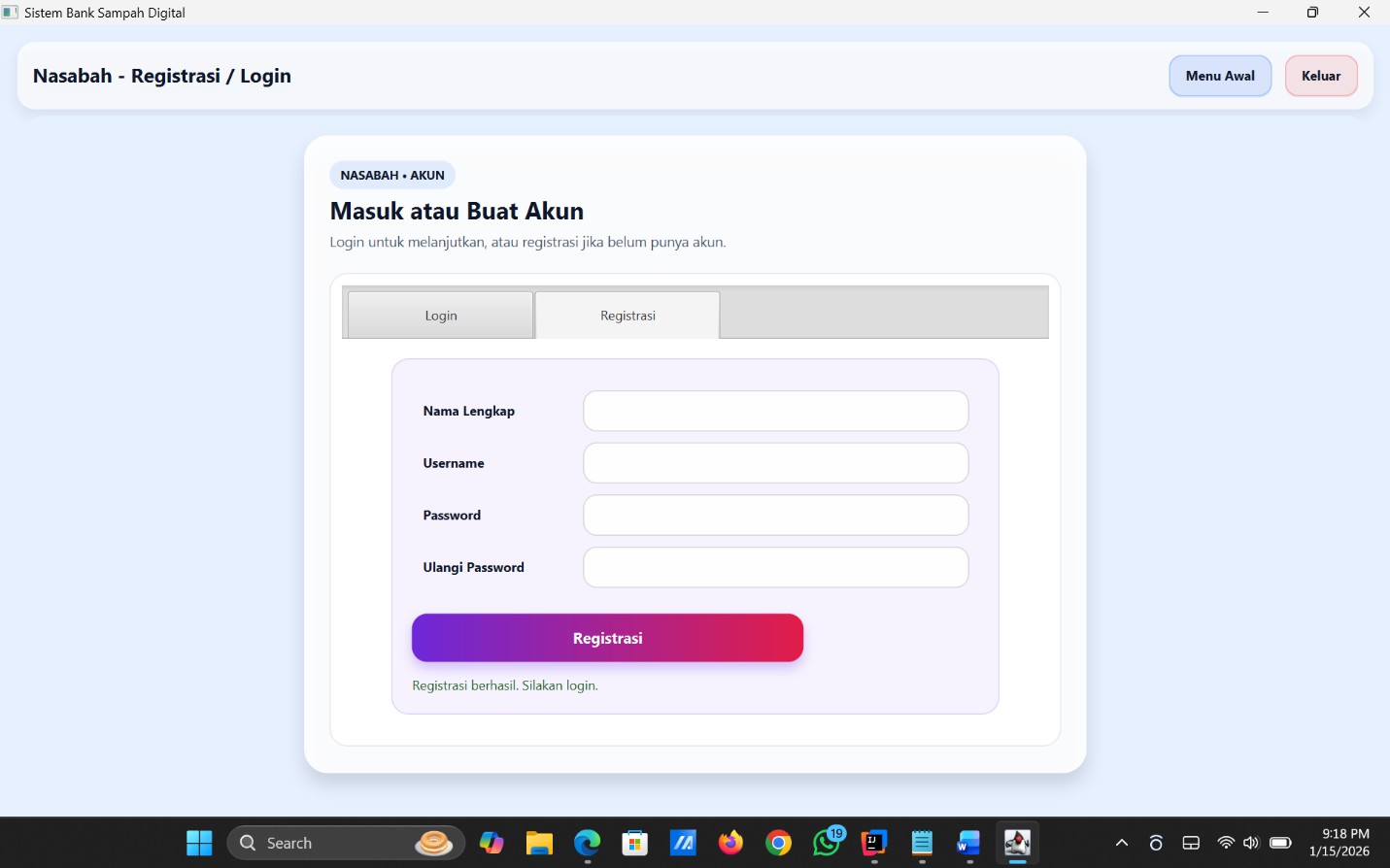
TransaksiSetor adalah turunan Transaksi yang merepresentasikan transaksi setor sampah. Kelas ini menyimpan detail tambahan berupa namaSampah, beratKg, dan hargaPerKg. Status transaksi setor dibuat SUCCESS karena transaksi setor dianggap valid ketika input lolos validasi.

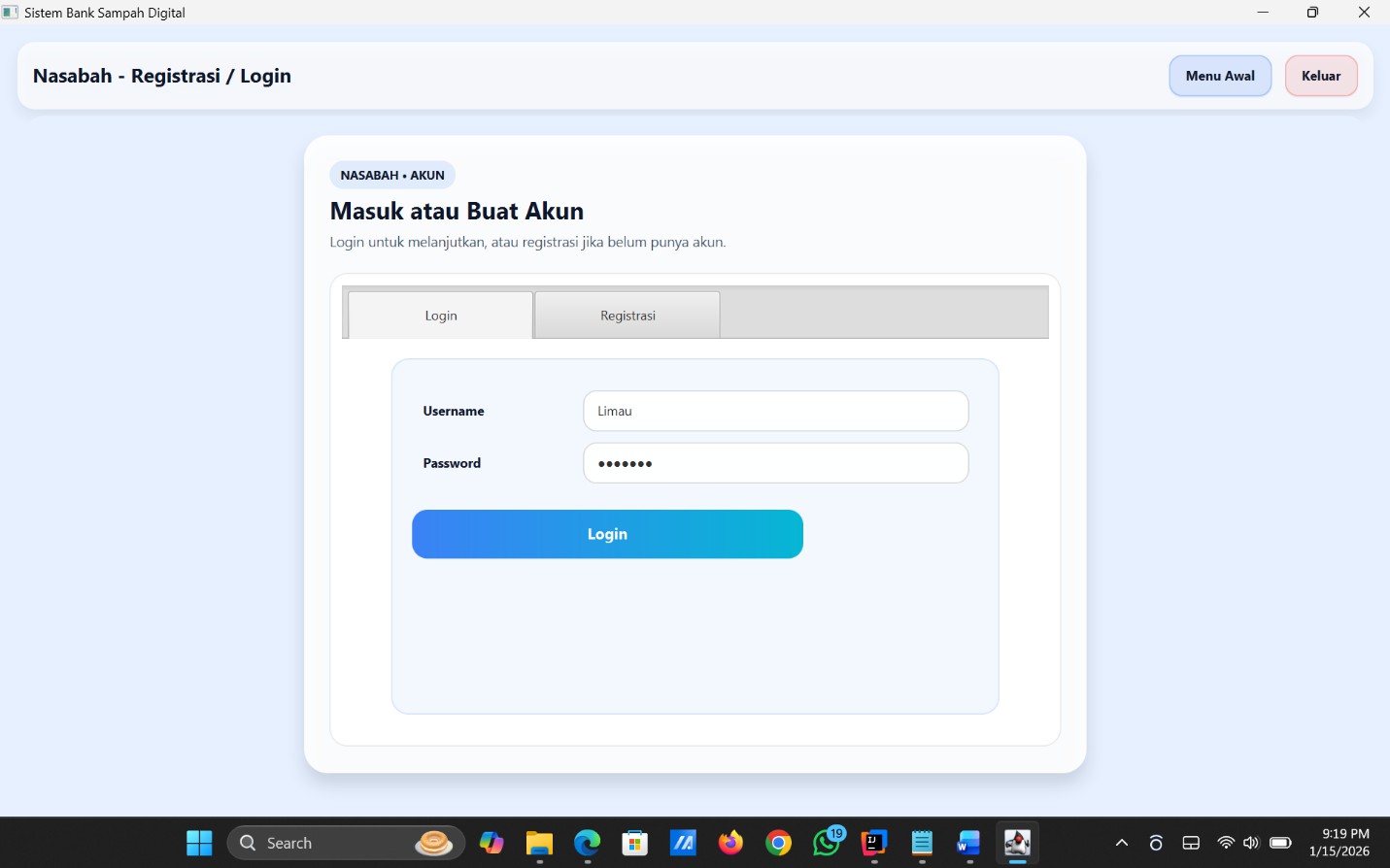
Total transaksi dihitung dari berat dikalikan harga/kg pada method total(), sedangkan method ringkasan() membentuk teks lengkap yang mencantumkan ID transaksi, jenis transaksi, nama sampah, perhitungan berat × harga, total rupiah, dan waktu. Ringkasan ini digunakan oleh riwayat nasabah agar bukti transaksi mudah dipahami tanpa membuka detail data internal.

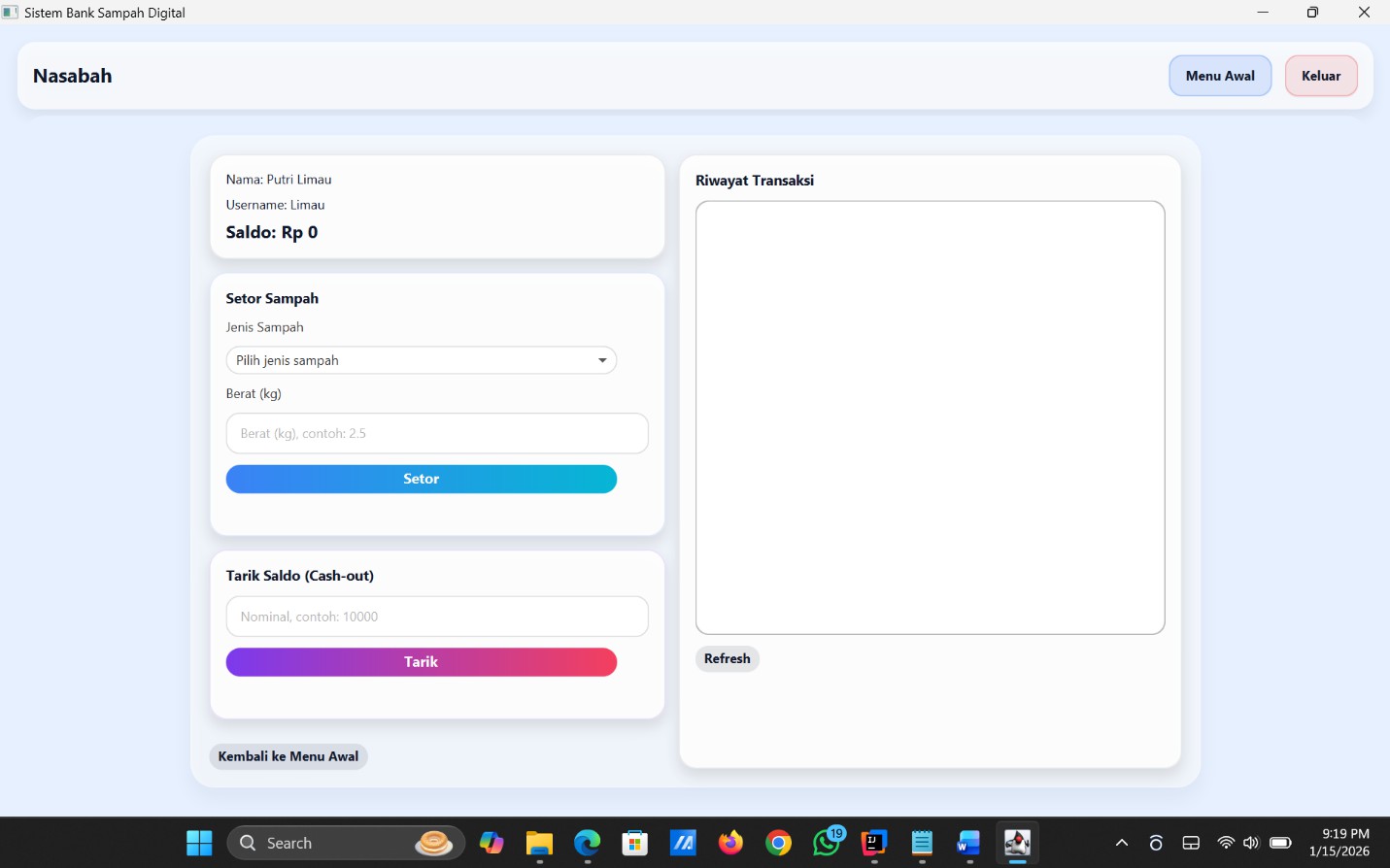
# TransaksiTarik.java

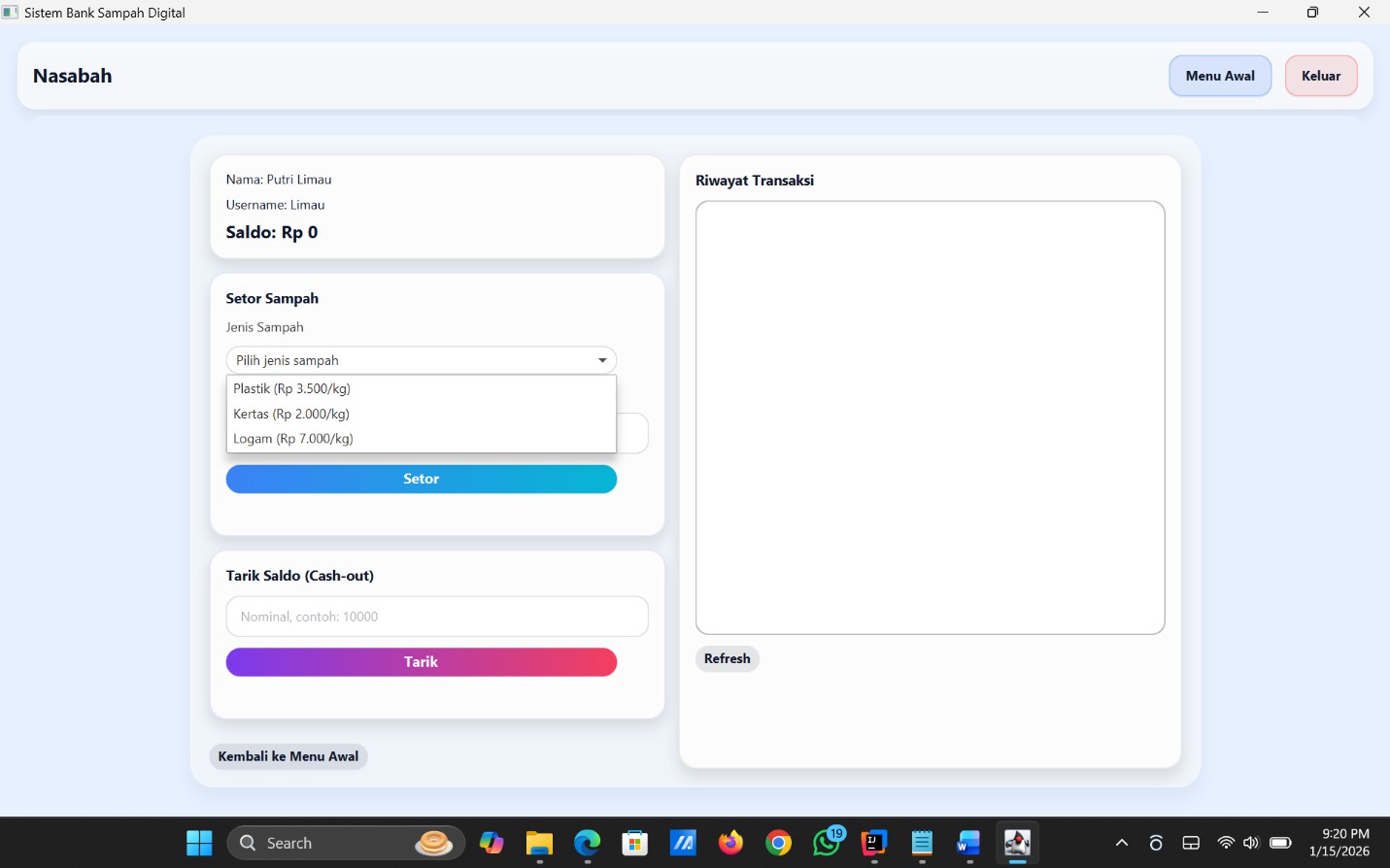
TransaksiTarik adalah turunan Transaksi yang merepresentasikan penarikan saldo. Kelas ini menyimpan nilai nominal tarik dan statusnya bisa SUCCESS atau FAILED tergantung hasil validasi saldo pada KioskService. Method total() mengembalikan nominal tarik, sementara method ringkasan() menghasilkan teks yang menampilkan ID transaksi, nominal rupiah, status keberhasilan (SUCCESS/FAILED), serta waktu transaksi. Dengan adanya status ini, riwayat nasabah tetap mencatat percobaan tarik yang gagal sebagai informasi yang transparan, dan operator dapat mengecualikan transaksi gagal dari perhitungan debit.

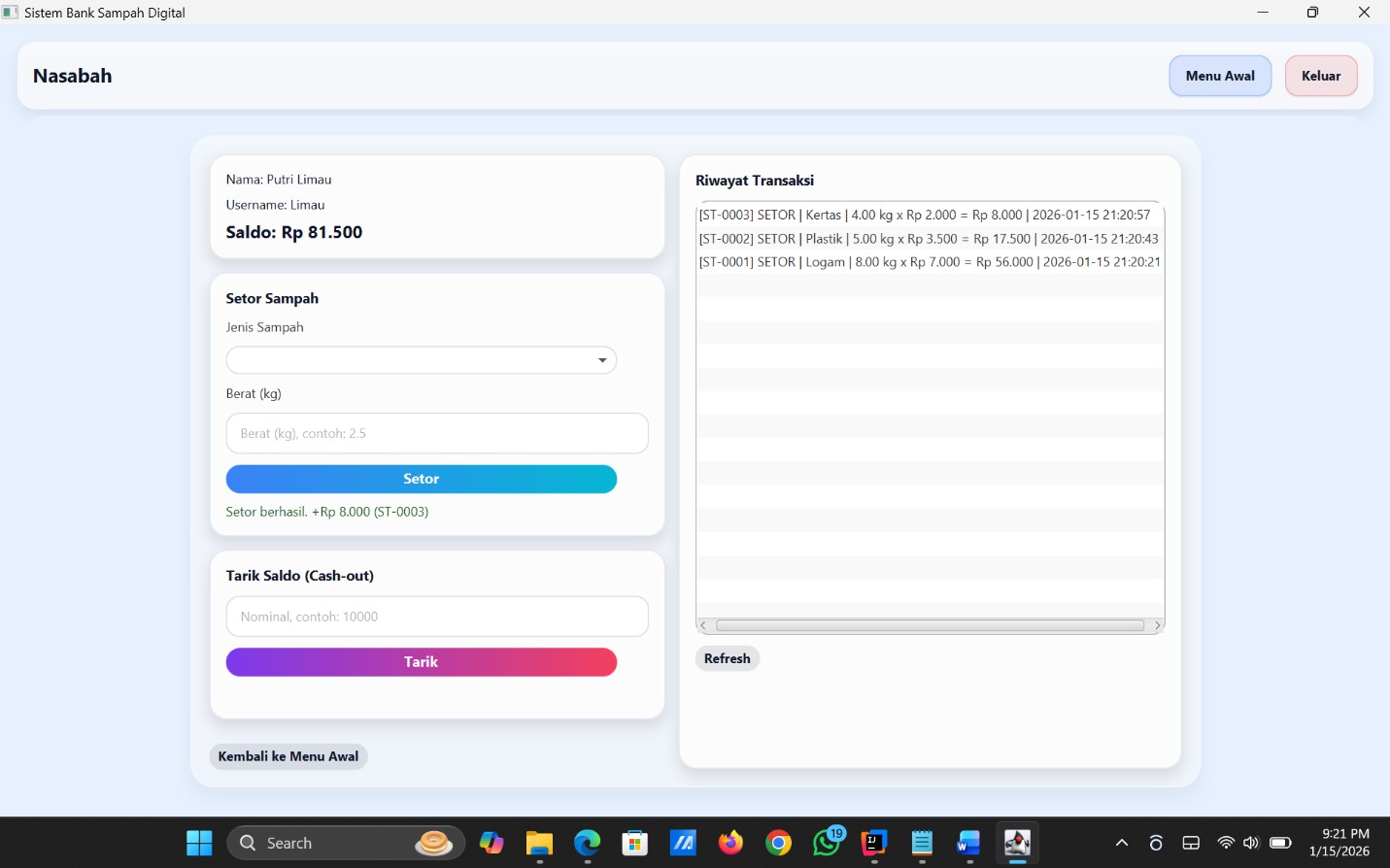
# Tampilan Akhir



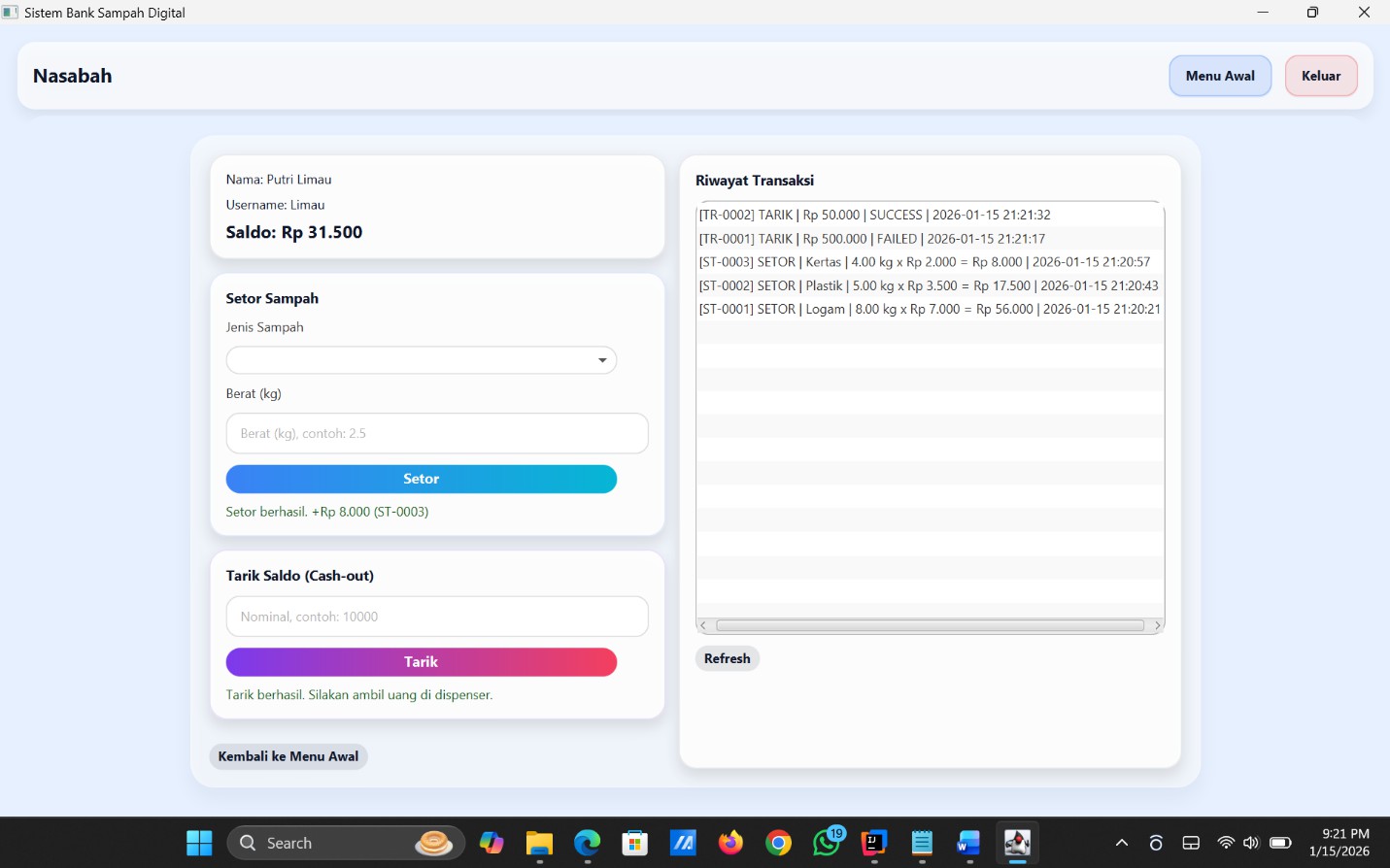
****

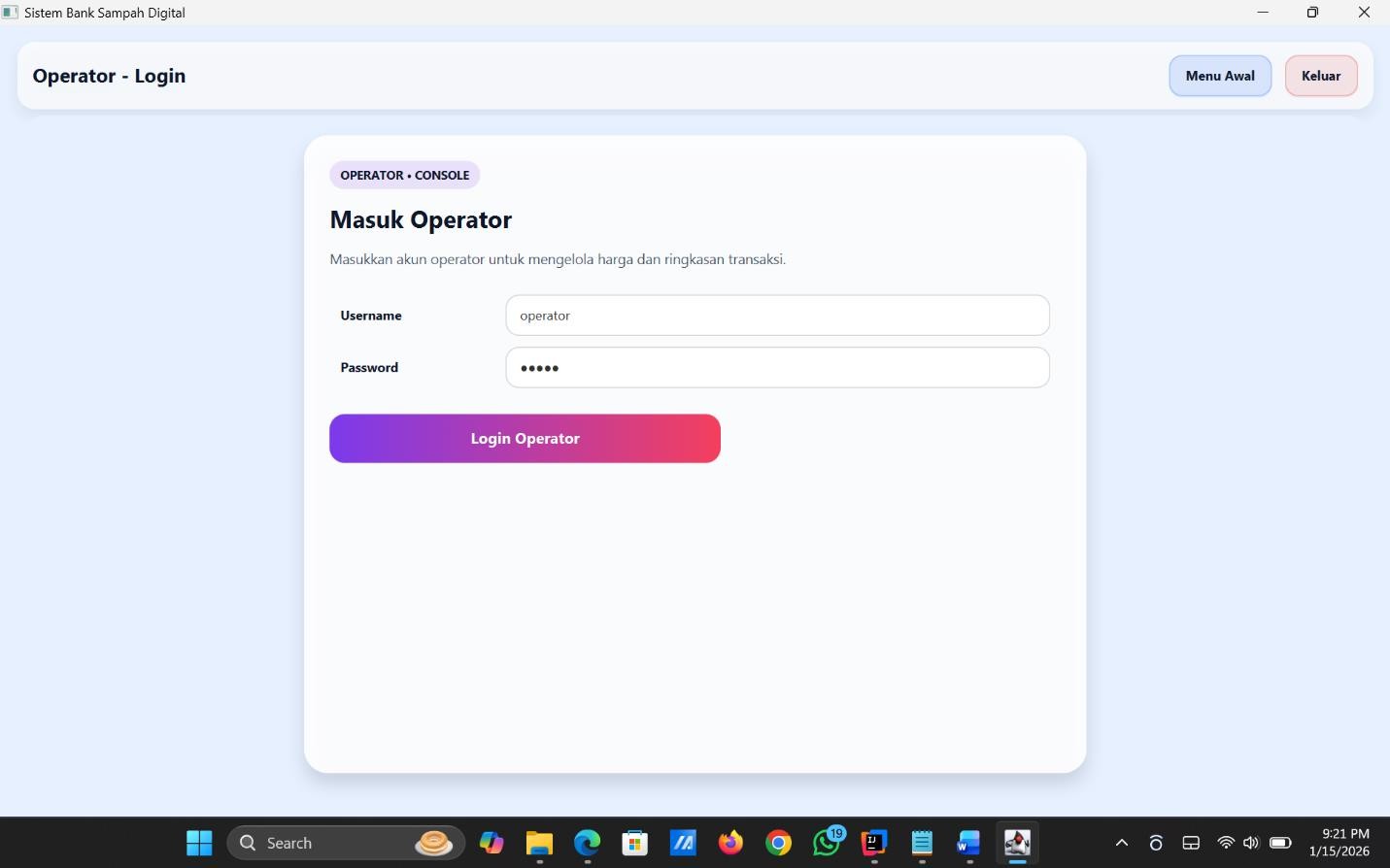


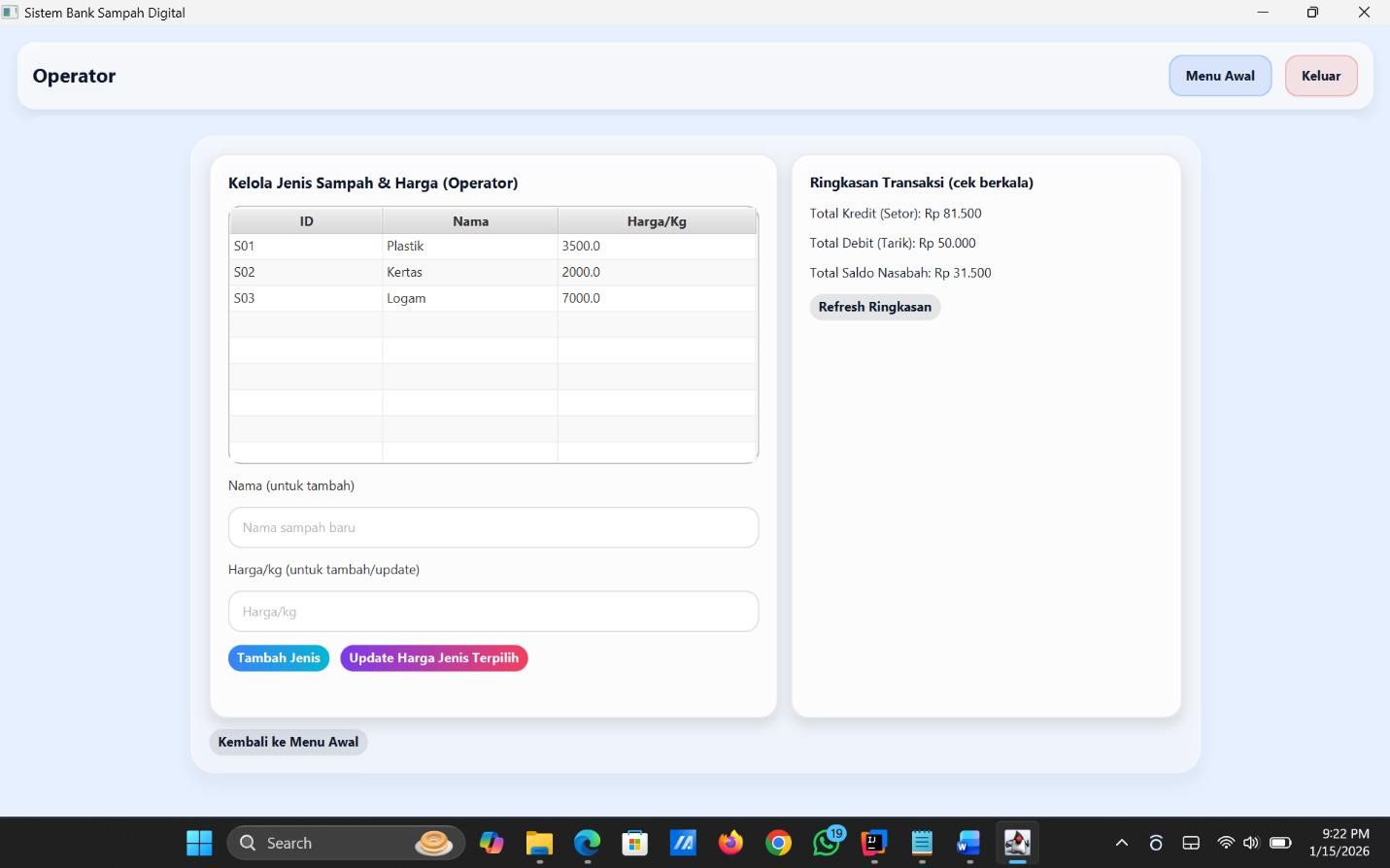
****



****



****



* + 1. **Tampilan Halaman Awal (Home)**

Pada tampilan awal aplikasi, pengguna disambut dengan halaman Selamat Datang yang berfungsi sebagai gerbang utama sistem. Halaman ini menampilkan judul aplikasi Sistem Bank Sampah Digital serta deskripsi singkat mengenai tujuan aplikasi. Di halaman ini juga tersedia dua tombol utama, yaitu Masuk sebagai Nasabah dan Masuk sebagai Operator.

# Tampilan Nasabah – Registrasi dan Login

Setelah memilih sebagai nasabah, pengguna diarahkan ke halaman Registrasi/Login Nasabah. Pada halaman ini tersedia dua tab, yaitu Login dan Registrasi, yang memungkinkan pengguna baru untuk membuat akun serta pengguna lama untuk masuk ke sistem.

Pada tab Registrasi, pengguna diminta mengisi data berupa nama lengkap, username, password, dan konfirmasi password. Sistem akan melakukan validasi input untuk memastikan data diisi dengan benar. Setelah registrasi berhasil, sistem menampilkan pesan konfirmasi yang menandakan bahwa akun telah dibuat dan pengguna dapat melanjutkan ke proses login.

Pada tab Login, pengguna memasukkan username dan password yang telah terdaftar. Jika data valid, sistem akan mengarahkan pengguna ke halaman utama nasabah. Jika tidak valid, sistem menampilkan pesan kesalahan sebagai umpan balik kepada pengguna.

# Tampilan Nasabah – Halaman Utama

Setelah login berhasil, nasabah akan masuk ke halaman utama nasabah. Halaman ini menjadi pusat aktivitas nasabah dan menampilkan beberapa komponen utama. Bagian pertama menampilkan informasi akun nasabah, yaitu nama, username, dan saldo terkini. Saldo akan diperbarui secara otomatis setiap kali terjadi transaksi setor maupun tarik.

Bagian Setor Sampah memungkinkan nasabah memilih jenis sampah dari daftar yang tersedia, seperti plastik, kertas, dan logam. Setiap jenis sampah memiliki harga per kilogram yang telah ditentukan oleh operator. Nasabah kemudian memasukkan berat sampah, dan sistem akan menghitung total nilai setor secara otomatis. Setelah proses setor berhasil, sistem menampilkan notifikasi keberhasilan beserta kode transaksi.

Bagian Tarik Saldo (Cash Out) digunakan untuk melakukan penarikan saldo. Nasabah memasukkan nominal penarikan, kemudian sistem akan memvalidasi apakah saldo mencukupi. Jika saldo tidak mencukupi, sistem menampilkan pesan gagal. Jika saldo mencukupi, saldo akan dikurangi dan sistem menampilkan pesan keberhasilan penarikan.

Di sisi kanan halaman, terdapat Riwayat Transaksi yang menampilkan seluruh aktivitas transaksi nasabah, baik setor maupun tarik. Setiap transaksi dicatat lengkap dengan jenis transaksi, jumlah, status (berhasil/gagal), serta waktu transaksi, sehingga nasabah dapat memantau aktivitasnya secara transparan.

# Tampilan Operator – Login

Untuk peran operator, sistem menyediakan halaman Login Operator yang terpisah dari nasabah. Operator diminta memasukkan username dan password khusus operator. Jika login berhasil, operator diarahkan ke halaman dashboard operator. Jika gagal, sistem akan memberikan notifikasi kesalahan.

# Tampilan Operator – Dashboard dan Pengelolaan Data

Pada halaman Dashboard Operator, terdiri dari dua bagian. Bagian pertama adalah Kelola Jenis Sampah dan Harga, yang menampilkan tabel berisi daftar jenis sampah beserta harga per kilogram. Operator dapat menambahkan jenis sampah baru serta memperbarui harga jenis sampah yang sudah ada. Perubahan harga ini akan langsung berdampak pada perhitungan setor sampah oleh nasabah.

Bagian kedua adalah Ringkasan Transaksi, yang menampilkan informasi agregat berupa total kredit (hasil setor), total debit (hasil tarik), serta total saldo nasabah. Ringkasan ini membantu operator dalam memantau kondisi sistem secara berkala.

# Kesimpulan

Berdasarkan hasil pengembangan yang telah dilakukan, dapat disimpulkan bahwa aplikasi Sistem Bank Sampah Digital berhasil diimplementasikan dengan baik dan sesuai dengan tujuan pembelajaran mata kuliah Pemrograman Berorientasi Objek. Aplikasi ini mampu mensimulasikan proses pengelolaan bank sampah secara digital dengan baik sesuai dengan desain awalnya. Perbedaan yang signifikan antara progress awal dan hasil akhir menunjukkan adanya peningkatan baik dari sisi desain antarmuka maupun fungsionalitas sistem.

# Tampilan Unggah GitHub

Screenshot tersebut menunjukkan status unggah skrip program pada repositori GitHub hingga mencapai tahap proyek final. Pada repositori tersebut terlihat seluruh file source code berupa class Java yang digunakan dalam pengembangan Sistem Bank Sampah Digital telah berhasil diunggah dan tersimpan dengan baik.

Selain skrip program, repositori juga memuat dokumen laporan yang mencerminkan tahapan pengerjaan proyek, mulai dari perencanaan, proses pengembangan, hingga laporan akhir. Hal ini menunjukkan bahwa proyek telah diselesaikan secara menyeluruh dan terdokumentasi dengan rapi menggunakan GitHub.

# Tampilan Unggah YouTube

**Link YouTube:** [**https://youtu.be/iCQAwiZZe2A?si=urFB1tlE\_GNHflGE**](https://youtu.be/iCQAwiZZe2A?si=urFB1tlE_GNHflGE)

Screenshot tersebut menunjukkan status publish video proyek Sistem Bank Sampah Digital pada platform YouTube. Terlihat bahwa video dengan judul **“PBO2025 – Tugas Projek – Sistem Bank Sampah Digital – Ali Tarmuji – Mutiara Avivah”** telah berhasil diunggah dan dipublikasikan di kanal milik pribadi, ditandai dengan tampilan halaman video yang dapat diakses secara publik.

Informasi jumlah penonton, tanggal unggah, serta keterangan judul dan deskripsi video menunjukkan bahwa video telah melalui proses unggah hingga tahap akhir dan siap ditonton oleh pengguna lain. Publikasi video ini berfungsi sebagai media dokumentasi dan presentasi hasil akhir proyek, sekaligus menjadi bukti bahwa sistem yang dikembangkan telah selesai dan dipresentasikan sesuai ketentuan tugas.

# Analisis Pengerjaan Projek

Analisis pengerjaan proyek Sistem Bank Sampah Digital ditinjau dari beberapa aspek utama, yaitu waktu, ketercapaian spesifikasi, biaya, kendala, serta tantangan ke depan. Dari sisi waktu, pengerjaan proyek dimulai pada hari Minggu sebagai tahap awal pengembangan program.

Proses implementasi utama sistem sebenarnya telah diselesaikan pada hari Jumat, namun dilakukan revisi tambahan pada bagian laporan sehingga penyelesaian akhir proyek berlangsung hingga hari Sabtu. Dengan demikian, keseluruhan pengerjaan proyek memerlukan waktu kurang lebih satu minggu dan dapat diselesaikan sesuai dengan batas waktu yang ditentukan.

Ditinjau dari ketercapaian spesifikasi, proyek ini telah memenuhi seluruh kebutuhan fungsional yang direncanakan sejak awal, bahkan melampaui ekspektasi awal pengembang. Sistem tidak hanya berjalan sesuai spesifikasi dasar, tetapi juga memiliki tampilan antarmuka yang lebih baik dan terstruktur, khususnya pada penerapan GUI berbasis JavaFX. Hal ini menunjukkan bahwa tujuan proyek dalam mengimplementasikan konsep pemrograman berorientasi objek dan antarmuka pengguna telah tercapai dengan baik.

Dari sisi biaya, pengerjaan proyek ini tidak memerlukan pengeluaran finansial karena seluruh proses pengembangan menggunakan perangkat lunak dan tools yang bersifat gratis dan tersedia secara bebas. Oleh karena itu, proyek ini dapat dikategorikan sebagai pengembangan sistem dengan biaya nol, yang efisien dari segi sumber daya.

Kendala utama yang dihadapi selama pengerjaan proyek terdapat pada tahap pemasangan dan konfigurasi GUI JavaFX di lingkungan pengembangan IntelliJ IDEA. Pada proses tersebut, JavaFX tidak dapat terhubung dengan baik karena adanya permasalahan pada pengaturan library dan modul, sehingga antarmuka grafis tidak dapat dijalankan secara langsung.

Ketidaksesuaian konfigurasi ini sempat menghambat proses pengembangan dan membutuhkan waktu tambahan untuk melakukan penyesuaian pada struktur modul dan dependensi proyek. Meskipun demikian, kendala tersebut akhirnya dapat diatasi setelah dilakukan pengaturan ulang hingga GUI JavaFX dapat berfungsi dengan baik sesuai kebutuhan sistem.

Adapun tantangan di masa depan meliputi pengembangan sistem agar lebih kompleks dan mendekati kondisi nyata, seperti integrasi dengan basis data permanen, peningkatan keamanan sistem, serta pengembangan fitur tambahan yang mendukung operasional bank sampah secara lebih luas. Dengan pengembangan lanjutan tersebut, sistem ini berpotensi menjadi aplikasi yang lebih aplikatif dan siap digunakan dalam skala yang lebih besar.