



## Sistem Grafic User Interface Aplikasi Berbasis Dekstop Penjualan Sederhana Menggunakan Java Netbeans

Mega Chrtistin Morys Lase, Vrenty Sri Berlian Nehe  
Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Nias Raya  
[megachristin@uniraya.ac.id](mailto:megachristin@uniraya.ac.id), [vrentysberliannehe@gmail.com](mailto:vrentysberliannehe@gmail.com)

### Abstrak

Perkembangan teknologi informasi telah memberikan pengaruh besar terhadap proses bisnis, termasuk dalam kegiatan penjualan yang sebelumnya dilakukan secara manual. Pada banyak usaha kecil dan menengah, proses transaksi seperti pencatatan barang, perhitungan total belanja, hingga pembuatan laporan penjualan masih dilakukan secara tradisional sehingga rawan kesalahan hitung, kurang efisien, dan memakan banyak waktu. Kondisi ini menunjukkan perlunya sebuah sistem penjualan sederhana yang mampu membantu pengguna dalam mengelola transaksi secara lebih cepat, akurat, dan terstruktur. Penelitian ini bertujuan merancang dan membangun aplikasi penjualan berbasis desktop menggunakan bahasa pemrograman Java pada NetBeans, dengan tampilan Graphical User Interface (GUI) yang dirancang sederhana dan mudah dipahami oleh pengguna. Metode yang digunakan dalam pengembangan aplikasi ini adalah model Waterfall yang terdiri dari analisis kebutuhan, perancangan sistem menggunakan UML, implementasi fitur aplikasi, dan pengujian menggunakan metode blackbox. Aplikasi yang dikembangkan memiliki beberapa fitur utama, antara lain input data barang, perhitungan subtotal dan total secara otomatis, penyimpanan transaksi, pembuatan laporan, serta kemampuan untuk mencetak struk pembelian. Hasil pengujian menunjukkan bahwa seluruh fungsi berjalan sesuai dengan kebutuhan, termasuk validasi input, perhitungan total, dan proses pencetakan struk. Aplikasi ini dapat membantu pengguna—khususnya pelaku usaha kecil—dalam mengelola transaksi penjualan sehingga lebih efisien, rapi, dan minim kesalahan. Dengan demikian, sistem yang dibangun terbukti mampu meningkatkan efektivitas proses penjualan dan berpotensi dikembangkan lebih lanjut dengan penambahan fitur seperti manajemen database dan pengelolaan stok barang.

Kata kunci: Aplikasi, Java, NetBeans, GUI, Waterfall

### 1. Latar Belakang

Perkembangan teknologi yang semakin maju membuat banyak kegiatan manusia beralih dari cara manual ke sistem yang lebih modern dan otomatis. Salah satu kegiatan yang sangat membutuhkan dukungan teknologi adalah proses penjualan. Pada banyak usaha kecil dan menengah, proses transaksi masih sering dilakukan dengan cara mencatat secara manual, menghitung harga satu per satu, dan menulis struk secara sederhana. Cara seperti ini tidak hanya memakan waktu, tetapi juga rawan kesalahan hitung, data bisa hilang, dan hasil perhitungan sering tidak akurat. Situasi ini membuat proses jual beli menjadi kurang efektif dan kurang praktis, terutama ketika jumlah transaksi meningkat.

Untuk menjawab masalah tersebut, diperlukan sebuah aplikasi penjualan yang sederhana, mudah digunakan, dan mampu membantu penjual dalam mengelola transaksi dengan lebih cepat dan rapi. Oleh karena itu, dibuatlah sebuah aplikasi penjualan berbasis Java menggunakan NetBeans sebagai tools pengembangan. Java dipilih karena memiliki kemampuan untuk membuat aplikasi GUI (Graphical User Interface) yang interaktif, stabil, dan dapat dijalankan pada berbagai perangkat komputer. Dengan bantuan NetBeans, proses pembuatan tampilan dan logika aplikasi menjadi lebih mudah dan terstruktur.

Aplikasi penjualan ini memiliki tampilan yang sederhana dan terdiri dari dua bagian utama. Pada bagian pertama terdapat form input data jualan yang berfungsi untuk memasukkan informasi barang. Form ini berisi kolom *Nama Barang*, *Harga Satuan (Rp)*, *Kuantitas*, dan *Potongan/Diskon (%)*. Setelah data diisi, pengguna dapat menekan tombol “Tambah ke Daftar” untuk memasukkan data ke tabel transaksi. Jika ingin membersihkan semua data input, tersedia tombol “Kosongin Daftar”. Bagian ini memudahkan penjual memasukkan barang tanpa harus menghitung manual, karena aplikasi akan memprosesnya secara otomatis.

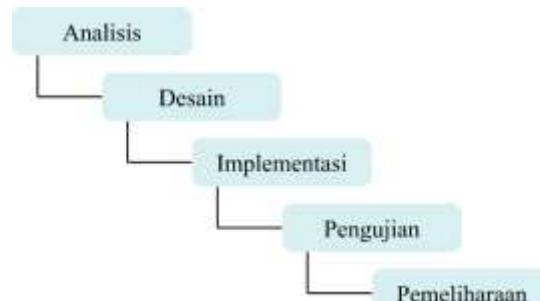
Selanjutnya, pada bagian kedua terdapat tabel Daftar Transaksi. Tabel ini menampilkan informasi transaksi secara lengkap, seperti *Waktu*, *Nama Barang*, *Harga*, *Qty*, *Diskon (%)*, dan total *Bayar*. Data yang tampil pada tabel ini adalah hasil dari input yang sebelumnya dimasukkan pada form. Pada bagian bawah tabel tersedia beberapa tombol penting, yaitu Cek Total untuk menghitung total keseluruhan belanja, Simpan Laporan untuk menyimpan hasil transaksi sebagai file laporan, Reset Semua untuk menghapus seluruh data transaksi, dan Cetak Struk untuk menghasilkan struk pembelian. Fitur cetak struk menjadi keunggulan utama karena memudahkan penjual memberikan bukti transaksi secara otomatis dan rapi kepada pelanggan.

Dengan adanya aplikasi ini, proses transaksi menjadi jauh lebih cepat, praktis, dan mengurangi risiko kesalahan hitung. Penjual tidak perlu lagi menulis struk secara manual, karena aplikasi sudah menyediakan fitur cetak otomatis. Selain itu, data transaksi tersimpan dengan baik dan dapat digunakan sebagai laporan penjualan. Aplikasi penjualan berbasis Java ini sangat bermanfaat bagi penjual kecil maupun menengah yang membutuhkan sistem sederhana namun efektif dalam mengelola transaksi harian.

## 2. Metode Penelitian

Dalam pengembangan Sistem Grafik User Interface (GUI) Aplikasi Penjualan Sederhana menggunakan Java NetBeans, metode Waterfall digunakan karena alurnya jelas, terstruktur, dan mudah diterapkan pada proyek berskala kecil. Metode Waterfall bekerja secara bertahap mulai dari tahap analisis, desain, implementasi, pengujian, hingga pemeliharaan. Pada tahap analisis, kebutuhan aplikasi dikumpulkan, seperti kebutuhan input data barang, perhitungan total, tampilan daftar transaksi, dan fitur cetak struk. Setelah kebutuhan jelas, tahap desain dilakukan untuk membuat rancangan tampilan GUI, menentukan letak form input, tabel transaksi, dan tombol-tombol seperti “Tambah ke Daftar”, “Cek Total”, dan “Cetak Struk”.

Setelah desain selesai, tahap implementasi dilakukan dengan membuat program menggunakan bahasa Java di NetBeans, mulai dari pembuatan form, logika perhitungan, hingga proses menampilkan data ke tabel. Tahap berikutnya adalah pengujian, yaitu mengecek apakah semua fitur berjalan dengan benar, seperti memastikan perhitungan total tidak salah, tombol bekerja sesuai fungsi, dan struk dapat dicetak dengan baik. Jika ditemukan kesalahan, dilakukan perbaikan. Tahap terakhir adalah pemeliharaan, yaitu memastikan aplikasi tetap bisa digunakan dengan baik, memperbaiki bug kecil, atau menambah fitur sesuai kebutuhan pengguna. Untuk lebih jelasnya mari perhatikan gambar 1.



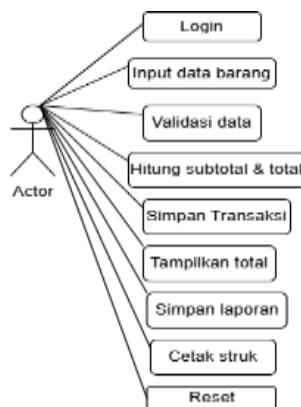
Gambar 1. Model Waterfall

### 2.1. Perancangan Sistem (UML)

Dalam pengembangan aplikasi penjualan sederhana ini, digunakan beberapa jenis diagram UML untuk menggambarkan cara kerja sistem secara lebih jelas. Use Case Diagram digunakan untuk menunjukkan hubungan antara pengguna (user) dan sistem, serta menjelaskan fitur apa saja yang dapat dilakukan pengguna, seperti memasukkan barang, menghitung total, menyimpan data, dan mencetak struk. Selanjutnya, Activity Diagram digunakan untuk menggambarkan alur aktivitas dari awal sampai akhir, misalnya proses ketika pengguna memasukkan data barang hingga sistem menampilkan hasil perhitungan dan mencetak struk. Diagram ini menggambarkan langkah kerja aplikasi secara runtut. Sementara itu, Sequence Diagram digunakan untuk menjelaskan urutan interaksi antara user interface, proses perhitungan, dan tampilan output. Diagram ini menunjukkan bagaimana perintah mengalir, seperti saat tombol “Tambah ke Daftar” ditekan, kemudian sistem memproses data dan menampilkannya di tabel.

### 2.1.1. Use Case Diagram

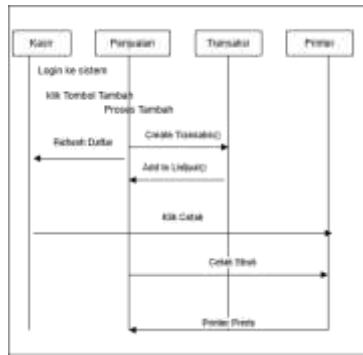
Use case pada aplikasi penjualan sederhana ini menggambarkan serangkaian aktivitas yang dapat dilakukan oleh pengguna saat menjalankan sistem. Proses dimulai dari Login, di mana pengguna harus masuk terlebih dahulu untuk dapat mengakses fitur aplikasi. Setelah berhasil login, pengguna dapat melakukan Input data barang, yaitu memasukkan informasi seperti nama barang, harga satuan, kuantitas, dan diskon. Data yang sudah dimasukkan kemudian melewati proses Validasi data untuk memastikan bahwa informasi yang diberikan lengkap dan tidak ada kesalahan input. Setelah data divalidasi, sistem akan melakukan Perhitungan subtotal dan total, yaitu menghitung nilai pembayaran berdasarkan jumlah barang, harga satuan, dan potongan yang diberikan. Jika perhitungan berhasil, pengguna dapat melakukan Simpan Transaksi, sehingga data transaksi tersebut masuk ke tabel transaksi. Selanjutnya, pengguna mampu melakukan Tampilkan total untuk melihat total keseluruhan pembelian yang sedang berlangsung. Aplikasi juga menyediakan fitur Simpan laporan, yang memungkinkan pengguna menyimpan seluruh transaksi dalam bentuk laporan harian atau bulanan. Selain itu, terdapat fitur Cetak struk yang digunakan untuk menghasilkan bukti transaksi secara otomatis sehingga pengguna bisa memberikan struk kepada pelanggan. Terakhir, tersedia use case Reset, yang berfungsi untuk menghapus semua data transaksi yang sedang ditampilkan sehingga aplikasi dapat digunakan kembali dari awal. Seluruh alur ini menunjukkan bagaimana pengguna berinteraksi dengan sistem secara bertahap, mulai dari login hingga mencetak struk, sesuai dengan proses bisnis aplikasi penjualan.



**Gambar 2.** Use Case Diagram

### 2.1.2. Sequence Diagram

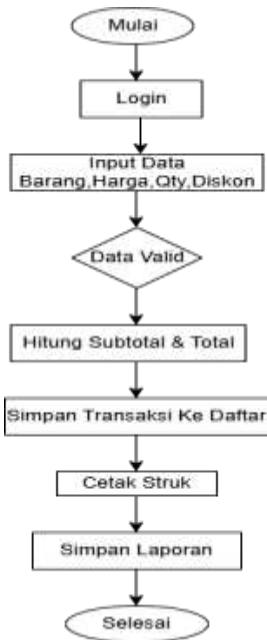
Sequence Diagram pada aplikasi penjualan sederhana ini menggambarkan alur interaksi antara empat objek utama, yaitu Kasir, Penjualan, Transaksi, dan Printer. Diagram ini menunjukkan bagaimana proses dijalankan secara berurutan mulai dari kasir melakukan login hingga struk berhasil dicetak. Proses diawali ketika Kasir login ke sistem. Setelah masuk, kasir melakukan aktivitas utama yaitu mengklik tombol "Tambah" pada menu penjualan untuk memasukkan barang ke daftar transaksi. Ketika tombol ini ditekan, objek Penjualan akan menjalankan proses *Tambah* untuk memvalidasi input barang dan menghitung subtotal ataupun total harga. Selanjutnya, objek Penjualan mengirimkan permintaan ke objek Transaksi untuk membuat data transaksi baru melalui operasi *Create Transaksi()*. Setelah transaksi berhasil dibuat, objek Transaksi mengirim proses kembali ke objek Penjualan dengan perintah *Add to ListJual()* sehingga data barang dapat ditambahkan ke tabel daftar transaksi pada tampilan aplikasi. Setelah data ditambahkan, sistem melakukan Refresh Daftar sehingga kasir dapat melihat daftar barang yang sudah diperbarui. Setelah semua barang dimasukkan, kasir menekan tombol Cetak, yang menjadi perintah untuk memulai proses pencetakan struk. Tombol ini membuat objek Penjualan berkomunikasi kembali dengan objek Transaksi untuk menghasilkan dokumen struk melalui proses *Cetak Struk*. Setelah struk berhasil diproses, objek Transaksi mengirimkan hasil tersebut ke Printer. Objek Printer kemudian menjalankan fungsi *Printer Prints*, yaitu proses mencetak struk secara fisik atau menampilkan hasil cetak secara digital. Setelah proses printing selesai, printer memberikan respon balik kepada Transaksi sebagai tanda bahwa struk sudah berhasil dicetak.



Gambar 3. Sequence Diagram

## 2.2. Flowchart

Proses dimulai dari tahap Mulai, kemudian pengguna melakukan Login ke dalam sistem untuk memastikan hanya pengguna yang berhak yang dapat mengakses aplikasi. Setelah berhasil masuk, pengguna dapat melakukan Input Data Barang, yaitu memasukkan nama barang, harga satuan, jumlah (qty), serta diskon bila diperlukan. Data yang telah dimasukkan kemudian akan melalui proses validasi untuk memastikan bahwa semua informasi yang diberikan sudah sesuai dan tidak ada kolom penting yang kosong atau salah isi. Jika data dinyatakan valid, sistem akan melanjutkan proses untuk menghitung subtotal dan total, termasuk perhitungan diskon bila ada. Setelah total dihitung, transaksi tersebut kemudian disimpan ke dalam daftar transaksi, sehingga data dapat muncul di tabel dan siap diproses lebih lanjut. Pengguna juga dapat melakukan cetak struk, yaitu menghasilkan bukti transaksi yang dapat diberikan kepada pelanggan. Selanjutnya, sistem menyediakan fitur simpan laporan untuk menyimpan seluruh transaksi ke dalam file yang dapat digunakan sebagai dokumentasi penjualan. Setelah seluruh proses selesai, flowchart diakhiri pada tahap Selesai. Alur ini menunjukkan bagaimana aplikasi bekerja secara berurutan, mulai dari input data hingga menghasilkan laporan dan struk secara otomatis, sehingga mempermudah proses transaksi bagi pengguna.



Gambar 4. Flowchart Aplikasi Kasir

## 2.3. Perangkat yang Digunakan

Dalam pembuatan aplikasi penjualan sederhana ini, perangkat yang digunakan terdiri dari perangkat lunak (software) dan perangkat keras (hardware) yang mendukung proses pengembangan. Dari sisi software, aplikasi

dibuat menggunakan bahasa pemrograman Java dengan bantuan NetBeans IDE sebagai tools utama untuk merancang tampilan GUI, menulis kode program, dan melakukan proses debugging. NetBeans dipilih karena memiliki fitur lengkap, mendukung pembuatan aplikasi berbasis Java Swing, serta memudahkan pengembangan dalam membuat class, mengatur form, dan mengelola seluruh struktur proyek. Sementara itu, untuk perangkat keras, pengembangan dilakukan menggunakan laptop ASUS dengan sistem operasi Windows 11, kapasitas RAM 8 GB, serta prosesor yang mendukung aktivitas pemrograman dan pengujian aplikasi. Laptop ini dilengkapi keyboard dan mouse atau touchpad sebagai perangkat input yang digunakan untuk mengetik kode, mengisi form, dan melakukan klik pada tombol-tombol di dalam aplikasi.

### 3. dan Pembahasan

#### 3.1 Implementasi Sistem

Implementasi sistem pada aplikasi penjualan sederhana berbasis Java NetBeans dilakukan dengan membangun seluruh fitur utama sesuai kebutuhan pengguna, mulai dari proses login hingga pencetakan struk. Tahap implementasi diawali dengan pembuatan tampilan antarmuka (GUI) menggunakan Java Swing, termasuk form login, form input barang, tabel daftar transaksi, serta tombol-tombol fungsi seperti Tambah, Reset, Simpan Laporan, Cek Total, dan Cetak Struk. Pada tahap ini, setiap komponen GUI dikaitkan dengan logika program melalui kelas-kelas Java yang menangani proses perhitungan subtotal dan total, validasi data, serta penyimpanan transaksi ke dalam daftar. Implementasi juga mencakup pembuatan fitur cetak struk yang mengubah data transaksi menjadi format struk yang dapat ditampilkan atau dicetak oleh pengguna. Selain itu, sistem dilengkapi fitur penyimpanan laporan transaksi agar pengguna dapat menyimpan hasil penjualan untuk kebutuhan dokumentasi. Setelah semua fitur selesai dibangun, dilakukan pengujian menggunakan metode BlackBox Testing untuk memastikan seluruh fungsi berjalan sesuai tujuan, seperti login berhasil, validasi input bekerja dengan benar, perhitungan total akurat, data transaksi tersimpan, dan struk berhasil dicetak tanpa error. Hasil implementasi ini menghasilkan aplikasi penjualan yang sederhana tetapi fungsional dan mampu mendukung proses transaksi harian dengan mudah.

##### 3.1.1 Tampilan Form Login

Tampilan login pada aplikasi kasir ini berfungsi sebagai pintu masuk utama sebelum pengguna dapat mengakses fitur penjualan. Pada tampilan tersebut, terdapat dua kolom input yaitu Username dan Password yang harus diisi oleh pengguna. Form ini dirancang untuk memastikan bahwa hanya pengguna yang berhak yang dapat mengoperasikan aplikasi, sehingga keamanan data transaksi dapat terjaga. Selain itu, terdapat dua tombol yaitu Login dan Batal. Tombol *Login* digunakan untuk memproses verifikasi data yang dimasukkan, sedangkan tombol *Batal* berfungsi untuk membatalkan proses dan menutup atau kembali ke halaman sebelumnya. Tampilan login ini dibuat dengan desain sederhana dan warna yang lembut agar mudah dipahami serta memberikan kenyamanan visual bagi pengguna dapat dilihat pada gambar 5.

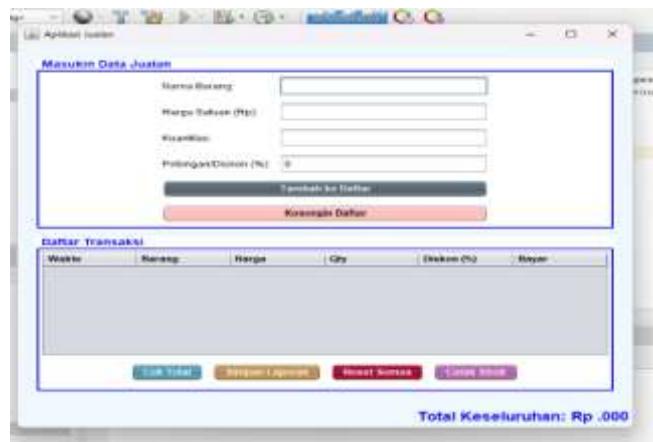


Gambar 5. Tampilan Login Aplikasi Kasir

##### 3.1.2 Tampilan Sistem

Tampilan pada gambar tersebut menunjukkan antarmuka utama dari Sistem Aplikasi Kasir, yang dirancang untuk memudahkan pengguna dalam memasukkan data penjualan serta mengelola transaksi secara praktis. Pada bagian atas terdapat form "Masukkan Data Jualan" yang berisi beberapa input, seperti nama barang, harga satuan, jumlah

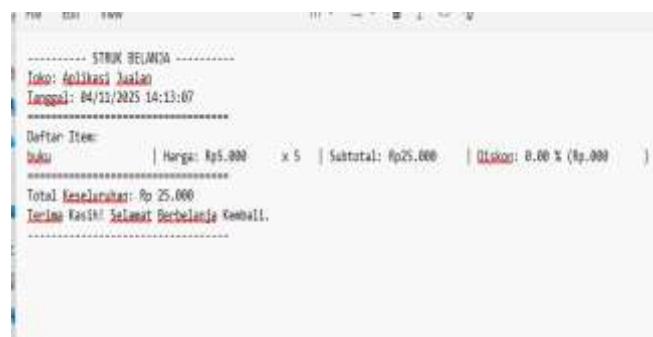
barang (kuantitas), dan potongan atau diskon. Setelah data diisi, pengguna dapat menambahkan transaksi ke tabel dengan menekan tombol “Tambah ke Daftar”, atau mengosongkan seluruh input menggunakan tombol “Kosongin Daftar”. Bagian bawah menampilkan tabel daftar transaksi yang berisi informasi waktu, nama barang, harga, jumlah (qty), diskon, dan total pembayaran. Di bawah tabel tersedia beberapa tombol fungsi, yaitu Cek Total, Simpan Laporan, Reset Semua, dan Cetak Struk, yang mempermudah proses pengelolaan transaksi dan pencetakan bukti pembelian. Pada pojok kanan bawah juga terdapat tampilan “Total Keseluruhan” yang menampilkan jumlah seluruh pembayaran secara otomatis. Secara keseluruhan, tampilan aplikasi ini dirancang sederhana, terstruktur, dan mudah digunakan dalam kegiatan transaksi penjualan dapat dilihat pada gambar 6 dibawah.



Gambar 6. Tampilan Sistem Aplikasi Kasir

### 3.1.3 Tampilan Struk

Gambar 7 dibawah ini menampilkan contoh struk belanja yang dihasilkan oleh Sistem Aplikasi Kasir setelah transaksi diproses. Struk ini disusun secara sederhana dan informatif, dimulai dengan judul “STRUK BELANJA” di bagian atas sebagai identitas dokumen. Di bawahnya tercantum informasi dasar seperti nama toko (“Aplikasi Jualan”) dan tanggal serta waktu transaksi, yang berfungsi sebagai bukti kapan transaksi dilakukan. Selanjutnya terdapat bagian Daftar Item, yang menampilkan rincian barang yang dibeli. Pada gambar, item yang dibeli adalah “buku”, dengan harga satuan Rp 5.000, jumlah pembelian sebanyak 5 unit, serta subtotal sebesar Rp 25.000. Informasi diskon juga ditampilkan dengan jelas, meskipun pada contoh ini diskon bernilai 0%. Bagian akhir struk menunjukkan Total Keseluruhan yang harus dibayar pelanggan, diikuti dengan ucapan terima kasih yang menambah kesan pelayanan ramah.



Gambar 7. Tampilan Struk Sistem Aplikasi Kasir

### 3.2 Pengujian Sistem (Blackbox)

Pengujian blackbox dilakukan untuk memastikan bahwa setiap fungsi dalam aplikasi penjualan bekerja sesuai dengan kebutuhan tanpa melihat kode program di dalamnya. Pada metode ini, pengujinya hanya fokus pada input yang dimasukkan dan output yang dihasilkan oleh sistem. Pengujian dilakukan pada beberapa fitur utama, seperti halaman login, input data barang, perhitungan subtotal dan total, penyimpanan transaksi, pencetakan struk, dan penyimpanan laporan. Tujuan dari pengujian ini adalah memastikan bahwa semua fungsi penting pada aplikasi

berjalan dengan benar, memberikan hasil yang sesuai, serta menampilkan pesan kesalahan yang tepat apabila terjadi kesalahan input dari pengguna. Hasil dari tabel pengujian menggunakan blackbox dapat dilihat pada tabel dibawah ini.

## 1. Pengujian Login

**Tabel 1.** Pengujian Fom Login Menggunakan Blacbox

No	Skenario Uji	Input	Output yang Diharapkan	Hasil
1	Username dan password benar	username: admin, password: admin	Berhasil login dan masuk ke menu utama	Berhasil
2	Username salah	username: salah, password: admin	Muncul pesan “Username atau Password salah”	Berhasil
3	Password salah	username: admin, password: salah	Muncul pesan “Username atau Password salah”	Berhasil
4	Field kosong	username: "", password: ""	Muncul pesan “Data tidak boleh kosong”	Berhasil

## 2. Pengujian Input Data Barang

**Tabel 2.** Pengujian Input Data Barang Menggunakan Blacbox

No	Skenario Uji	Input	Output yang Diharapkan	Hasil
1	Mengisi semua data dengan benar	Barang: Rokok, Harga: 20000, Qty: 2, Diskon: 0	Data masuk ke tabel	Berhasil
2	Tidak mengisi salah satu field	Barang kosong, harga terisi	Muncul pesan “Data belum lengkap”	Berhasil
3	Mengisi harga atau qty dengan huruf	Harga: ABC	Muncul pesan “Input harus angka”	Berhasil

## 3. Pengujian Perhitungan Subtotal & Total

**Tabel 3.** Pengujian Operasi Perhitungan Menggunakan Blacbox

No	Skenario Uji	Input	Output yang Diharapkan	Hasil
1	Hitung subtotal tanpa diskon	Harga: 10000, Qty: 3	Subtotal = 30000	Berhasil
2	Hitung subtotal dengan diskon	Harga: 20000, Qty: 2, Diskon: 10%	Total = 36000	Berhasil
3	Diskon lebih dari 100%	Diskon: 120	Muncul pesan “Diskon tidak valid”	Berhasil

## 4. Pengujian Penyimpanan Transaksi

**Tabel 4.** Pengujian Penyimpanan Transaksi Menggunakan Blacbox

No	Skenario Uji	Input	Output yang Diharapkan	Hasil
1	Tambah transaksi valid	Input lengkap	Tersimpan ke daftar tabel	Berhasil
2	Input kosong	Tidak mengisi barang	Muncul pesan kesalahan	Berhasil

## 5. Pengujian Cetak Struk

**Tabel 5.** Pengujian Cetak Struk Menggunakan Blacbox

No	Skenario Uji	Input	Output yang Diharapkan	Hasil
1	Klik tombol Cetak	Data transaksi ada	Struk tampil atau tercetak	Berhasil
2	Klik tombol Cetak tanpa data	Tidak ada transaksi	Muncul pesan “Tidak ada data untuk dicetak”	Berhasil

## 6. Pengujian Simpan Laporan

**Tabel 6.** Pengujian Simpan Laporan Menggunakan Blacbox

No	Skenario Uji	Input	Output yang Diharapkan	Hasil
1	Simpan laporan	Ada transaksi	Laporan berhasil dibuat	Berhasil
2	Simpan tanpa transaksi	Data kosong	Muncul pesan “Tidak ada data untuk disimpan”	Berhasil

#### 4. Kesimpulan

Berdasarkan hasil pengembangan dan pengujian, aplikasi penjualan sederhana berbasis Java NetBeans ini berhasil dibuat dan dapat bekerja sesuai kebutuhan. Aplikasi ini mampu membantu pengguna dalam mengelola transaksi dengan lebih cepat, rapi, dan tanpa perlu menghitung secara manual. Seluruh fitur utama seperti login, input data barang, perhitungan subtotal dan total, penyimpanan transaksi, pencetakan struk, serta simpan laporan dapat berjalan dengan baik berdasarkan hasil pengujian blackbox. Aplikasi ini membuktikan bahwa sistem sederhana sekalipun dapat meningkatkan efektivitas dan mengurangi kesalahan dalam proses penjualan, terutama bagi usaha kecil dan menengah. Ke depannya, aplikasi ini masih dapat dikembangkan lagi, misalnya dengan penambahan database, fitur stok barang, atau integrasi sistem yang lebih lengkap.

#### Referensi

1. Deitel, P., & Deitel, H. (2019). *Java: How to Program*. Pearson Education.
2. Eko, B. (2020). *Pemrograman Berorientasi Objek dengan Java*. Informatika Bandung.
3. Halim, A. (2021). *Membangun Aplikasi Desktop Menggunakan Java NetBeans*. Andi Publisher.
4. Kadir, A. (2018). *Dasar-Dasar Pemrograman Java*. Andi.
5. Kristanto, A. (2018). *Perancangan Sistem Informasi dan Aplikasinya*. Gava Media.
6. Nugroho, A. (2017). *Rekayasa Perangkat Lunak Berorientasi Objek*. Informatika.
7. Pratama, F. R. (2020). "Pengembangan Aplikasi Kasir Berbasis Desktop Menggunakan Java dan NetBeans." *Jurnal Teknologi Informasi*, 8(2), 55–62.
8. Siregar, J. (2021). *Pengenalan UML untuk Perancangan Sistem*. Graha Ilmu.
9. Sutarman. (2017). *Pengantar Teknologi Informasi*. Bumi Aksara.
10. Wahana Komputer. (2019). *Belajar Pemrograman Java untuk Pemula*. Andi Publisher.