

**LAPORAN PROYEK AKHIR PRAKTIKUM**  
**MATA KULIAH ALGORITMA DAN PEMROGRAMAN DASAR**



**DATA MANAJEMEN KEUANGAN SUATU PERUSAHAAN**

**Oleh:**

**Kelompok 2**

**M. NABIL RAHMATULLAH**  
**ANINDTYA PUJI ASTARI**  
**RIDWAN NUR RAHMAN**  
**NOU JULYANAH MAZUWA**

**2409106046**  
**2409106063**  
**2409106064**  
**2409106066**

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA 2024**

## KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Allah SWT, atas rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan laporan proyek akhir program “Manajemen Keuangan Suatu Perusahaan” ini dengan baik.

Program yang kami kembangkan bertujuan untuk membantu mengelola data keuangan perusahaan retail secara efisien. Program ini memuat fitur CRUD (*Create, Read, Update, Delete*) dengan dua jenis pengguna, yaitu admin (supervisor) yang memiliki akses penuh terhadap data dan sales yang memiliki akses terbatas. Dengan pendekatan ini, kami berharap program dapat mendukung pengambilan keputusan yang lebih cepat dan akurat dalam pengelolaan keuangan perusahaan retail.

Selama proses penyusunan program dan laporan, kami menghadapi berbagai kendala, seperti *debugging* program, pengelolaan data, dan sinkronisasi fitur antar pengguna. Namun, berkat bimbingan dari dosen pengampu, bantuan asisten laboratorium, serta kerja sama dengan rekan-rekan, semua tantangan tersebut dapat kami atasi dengan baik. Untuk itu, kami mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan dalam proses ini.

Kami menyadari bahwa laporan ini masih jauh dari sempurna, baik dari segi isi maupun penyajiannya. Oleh karena itu, kami terbuka terhadap kritik dan saran yang membangun. Semoga laporan ini dapat memberikan manfaat bagi pembaca, khususnya dalam memahami penerapan teknologi Python untuk manajemen keuangan perusahaan.

Samarinda, 20 November 2024

## TAKARIR

Daftar padanan kata bahasa asing dalam bahasa Indonesia yang digunakan adalah sebagai berikut:

<i>Input</i>	Memasukkan
<i>Login</i>	Masuk
<i>Register</i>	Daftar
<i>Logout</i>	Keluar
<i>Bonus</i>	Insentif
<i>Target</i>	Sasaran yang ditetapkan untuk dicapai
<i>Visualization</i>	Visualisasi
<i>Create</i>	Membuat
<i>Read</i>	Membaca
<i>Update</i>	Memperbarui
<i>Delete</i>	Menghapus
<i>Debugging</i>	Melakukan Bug
<i>Real-time</i>	Waktu Terkini
<i>User-friendly</i>	Mudah Dipahami
<i>Username</i>	Nama Pengguna
<i>Password</i>	Kata Sandi
<i>Role</i>	Peran
<i>User</i>	Pengguna

## DAFTAR ISI

KATA PENGANTAR .....	ii
TAKARIR.....	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR .....	v
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Kebutuhan Fungsional.....	2
1.3 Rumusan Masalah .....	3
1.4 Batasan Masalah.....	3
1.5 Tujuan.....	3
BAB II PERANCANGAN.....	4
2.1 Analisis Program .....	4
2.2 Flowchart.....	6
2.3 Konsep/Materi Praktikum yang dipakai.....	11
BAB III HASIL DAN PEMBAHASAN.....	13
3.1 Tampilan Program .....	13
3.2 Source Code .....	25
BAB IV PENUTUP .....	42
4.1 Kesimpulan.....	42
4.2 Saran .....	42
DAFTAR PUSTAKA .....	43
LAMPIRAN.....	44

## DAFTAR GAMBAR

Gambar 2. 3 Flowchart menu sales .....	10
Gambar 3. 1.....	13
Gambar 3. 2 .....	13
Gambar 3. 3 .....	14
Gambar 3. 4 .....	14
Gambar 3. 5 .....	15
Gambar 3. 6 .....	15
Gambar 3. 7 .....	16
Gambar 3. 8 .....	16
Gambar 3. 9 .....	17
Gambar 3. 10 .....	17
Gambar 3. 11 .....	18
Gambar 3. 12 .....	18
Gambar 3. 13 .....	19
Gambar 3. 14 .....	19
Gambar 3. 15 .....	20
Gambar 3. 16 .....	20
Gambar 3. 17 .....	21
Gambar 3. 18 .....	21
Gambar 3. 19 .....	22
Gambar 3. 20 .....	23
Gambar 3. 21 .....	24
Gambar 3. 22 .....	25
Gambar 3. 23 .....	26
Gambar 3. 24 .....	27
Gambar 3. 25 .....	28
Gambar 3. 26 .....	29
Gambar 3. 27 .....	30
Gambar 3. 28 .....	31
Gambar 3. 29 .....	32
Gambar 3. 30 .....	33
Gambar 3. 31 .....	33
Gambar 3. 32 .....	34
Gambar 3. 33 .....	34
Gambar 3. 34 .....	35

**Gambar 3. 35** ..... 36

**Gambar 3. 36** ..... 37

**Gambar 3. 37** ..... 37

**Gambar 3. 38** ..... 38

**Gambar 3. 39** ..... 39

**Gambar 3. 40** ..... 40

**Gambar 3. 41** ..... 41

**Gambar 4. 2**..... 44

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Manajemen keuangan adalah hal yang sangat penting dalam menjalankan sebuah perusahaan. Keberhasilan bisnis sangat dipengaruhi oleh bagaimana perusahaan mengelola uang dan sumber daya keuangannya dengan baik, tepat, dan terbuka. Namun, banyak perusahaan yang masih mengalami kesulitan dalam mengelola keuangan, terutama jika mereka masih menggunakan cara manual atau sistem yang tidak terhubung satu sama lain. Hal ini bisa menyebabkan berbagai masalah, seperti kesalahan pencatatan, kesulitan dalam memantau kondisi keuangan secara *real-time*, dan ketidakefisienan dalam pengelolaan data keuangan.

Oleh karena itu, kami merancang sebuah program data manajemen keuangan suatu perusahaan yang akan membantu dalam mengelola data keuangan suatu perusahaan. Program ini akan dilengkapi dengan berbagai fitur mempermudah proses *login* dan registrasi pengguna, memberikan perbedaan hak akses yang jelas antara admin dan staf penjualan, menyediakan fitur manajemen data yang efisien, termasuk pengolahan data transaksi dan administrasi. dan meningkatkan efisiensi kerja dengan antarmuka yang ramah pengguna (*user-friendly*).

Dalam laporan ini, kami akan menjelaskan secara rinci mengenai konsep dan desain dari program data manajemen keuangan suatu perusahaan yang kami rancang, serta manfaat dan tujuan dari penggunaan program ini. Kami berharap laporan ini dapat menjadi panduan bagi pihak-pihak terkait untuk memahami konsep dan manfaat dari program data manajemen keuangan suatu perusahaan yang kami rancang.

## 1.2 Kebutuhan Fungsional

Pada awal program ini terdapat 3 menu yang tersedia, yaitu :

1. *Login*: Pengguna bisa masuk ke sistem dengan memasukkan *username* dan *password* yang benar sebagai admin atau sales.
2. Registrasi: Pengguna yang belum memiliki akun dapat membuat akun baru agar bisa menggunakan sistem.
3. *Logout*: Pengguna diberikan pilihan untuk keluar dari aplikasi dan menutup sesi penggunaan sistem.

Jika *login* sebagai admin maka akan menampilkan menu berikut :

1. Lihat produk : Membantu admin melihat data produk dengan mudah sebelum melakukan tindakan seperti menambah, mengedit, atau menghapus produk.
2. Status performa sales : Memungkinkan admin untuk memantau dan mengelola kinerja sales dengan mudah dan efisien.
3. Lihat penjualan : Memudahkan admin untuk memantau transaksi dan mengetahui produk yang paling banyak terjual.
4. Lihat keuangan : Mempermudah admin untuk melihat dan memahami kondisi keuangan perusahaan secara lengkap tanpa perlu menghitung secara manual.
5. *Visualization* data : Memudahkan admin dalam menganalisis pola penjualan untuk membantu mengambil keputusan strategis.
6. *Logout* : menu ini untuk digunakan admin untuk keluar dari akunnya.

Jika *login* sebagai sales maka akan menampilkan menu berikut :

1. Lihat produk : Menu ini membantu sales untuk melihat produk yang tersedia dan memastikan apakah stok cukup untuk dijual.
2. Lihat penjualan : Sales bisa mengecek kembali transaksi yang sudah dilakukan dan memantau kinerja penjualan mereka.



3. Lihat performa : Membantu sales melihat apakah mereka sudah mencapai *target* penjualannya atau belum, dan seberapa dekat mereka dengan *bonus* yang dijanjikan.
4. Jual produk : Mempermudah sales untuk mencatat transaksi penjualan, mengurangi stok produk, dan langsung mendapatkan informasi harga total.
5. *Logout* : Menu ini memungkinkan sales untuk keluar dari sesi mereka dan kembali ke layar utama.

### **1.3 Rumusan Masalah**

Berdasarkan studi kasus dari deskripsi masalah diatas, maka kita dapat mengetahui rumusan masalah ini adalah:

1. Bagaimana cara mengelola data produk dengan mudah?
2. Bagaimana memantau performa penjualan sales dan pencapaian *target*?
3. Bagaimana membuat laporan keuangan dan visualisasi data penjualan?

### **1.4 Batasan Masalah**

1. Hanya Ada Dua Jenis Pengguna (Admin dan Sales)
2. Fitur Terbatas pada Pengelolaan Produk dan Penjualan
3. Analisis Keuangan Terbatas
4. Sales Hanya Bisa Melihat Penjualannya Sendiri

### **1.5 Tujuan**

1. Membangun sistem untuk mempermudah manajemen produk, penjualan, dan performa sales.
2. Menyediakan laporan keuangan dan performa secara *real-time*.
3. Memberikan kemudahan akses bagi admin dan sales.
4. Menyediakan visualisasi data penjualan untuk membantu analisis bisnis.

## **BAB II**

### **PERANCANGAN**

#### **2.1 Analisis Program**

Program dibagi menjadi beberapa modul utama:

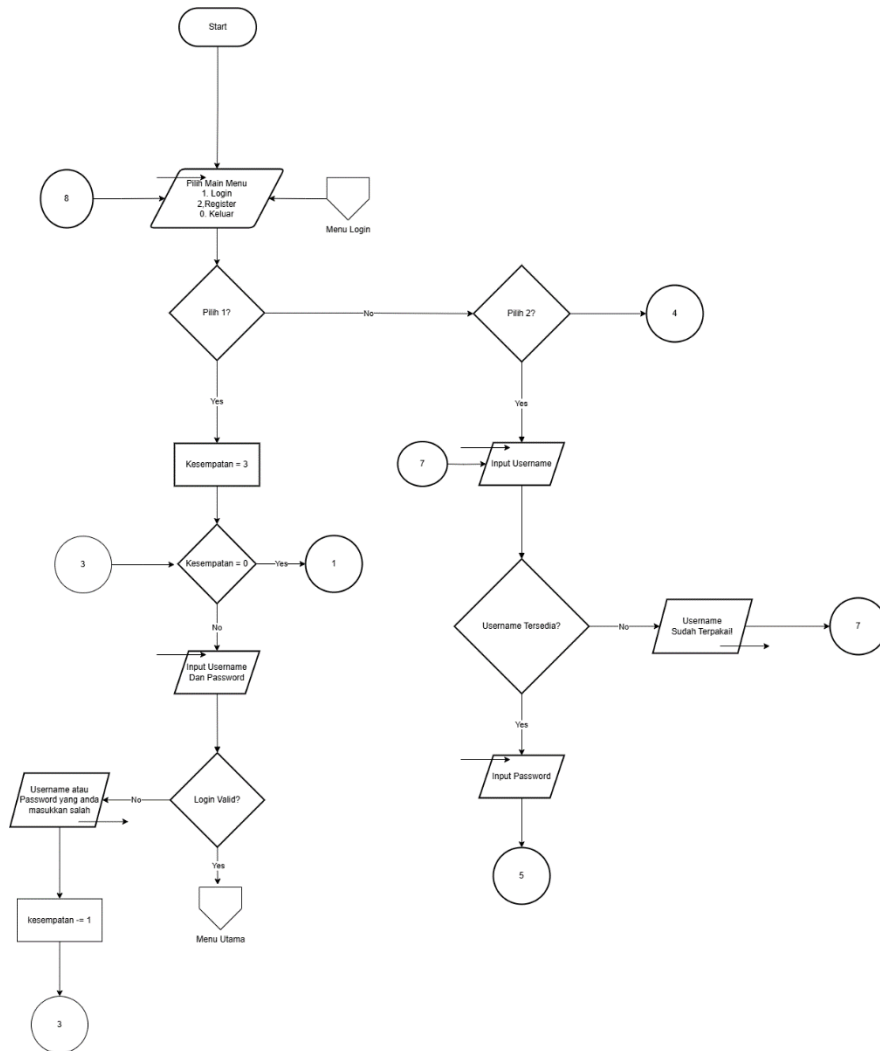
1. Auth.py : Modul untuk registrasi dan *login*.
2. Admin.py : Mengelola produk, penjualan, performa sales, laporan keuangan, dan *Visualization*.
3. Sales.py : Digunakan oleh sales untuk melihat produk, melakukan transaksi, dan melihat performa pribadi.
4. Main.py : Modul utama untuk menjalankan aplikasi.
5. Product.csv : Menyimpan data produk.
6. Sales.csv : Menyimpan data sales dan performance sales.
7. Users.csv : Menyimpan data registrasi *username*, *password*, dan *role user*.
8. Transaction.csv : Menyimpan data transaksi penjualan.

Program ini adalah sistem data manajemen keuangan suatu perusahaan. Ketika dijalankan, program akan menampilkan menu utama dengan 3 pilihan:

1. *Login*: Untuk masuk ke sistem. Setelah *login*, pengguna akan diarahkan ke menu khusus sesuai dengan perannya, yaitu:
  - 1) Admin: Jika pengguna adalah seorang admin, mereka akan masuk ke menu admin.
  - 2) Sales: Jika pengguna adalah seorang sales, mereka akan masuk ke menu sales.
2. *Register*: Untuk mendaftar akun baru. Jika belum punya akun, pengguna bisa mendaftarkan diri melalui opsi ini.
3. Keluar: Jika pengguna memilih opsi ini, program akan berhenti.

4. Tampilkan Menu Utama: Saat pertama kali dijalankan, program akan membersihkan layar dan menampilkan menu utama.
5. Pengguna Memilih Opsi:
6. Jika memilih “1” (*Login*):
  - a. Program akan meminta pengguna memasukkan *username* dan *password*.
  - b. Jika berhasil *login*:
    - i. Pengguna admin akan masuk ke menu admin.
    - ii. Pengguna sales akan masuk ke menu sales.
  - c. Jika gagal *login*, pengguna akan diminta mencoba lagi.
7. Jika memilih “2” (*Register*):
  - a. Program akan memandu pengguna untuk membuat akun baru.
  - b. Setelah selesai, pengguna akan kembali ke menu utama.
8. Jika memilih “0” (*Keluar*):
  - a. Program menampilkan pesan terima kasih, lalu berhenti.
9. Jika Input Tidak Valid:
10. Program akan menampilkan pesan kesalahan (misalnya, "Menu tidak valid!") dan meminta pengguna menekan Enter untuk kembali ke menu utama.
11. Program Berjalan Terus:
12. Setelah setiap aksi (*login*, *register*, atau kesalahan), program kembali ke menu utama, kecuali pengguna memilih untuk keluar.

## 2.2 Flowchart

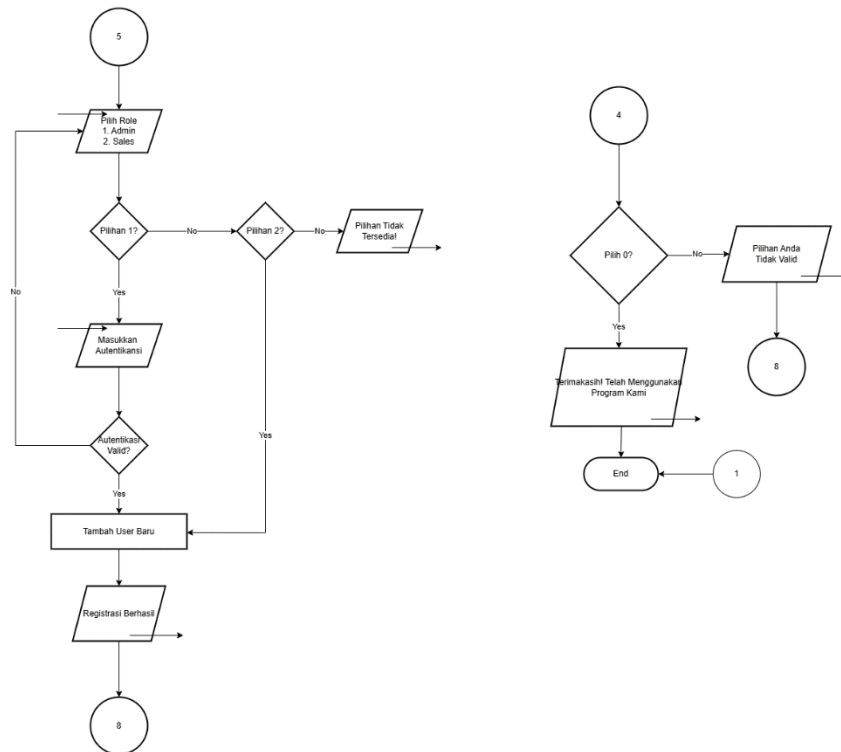


**Gambar 2 1**Flowchart menu login

Program ini adalah sistem manajemen retail yang dimulai dengan menampilkan menu utama yang terdiri dari tiga pilihan: *Login*, *Register*, dan *Keluar*.

Jika pengguna memilih *Login*, mereka akan diminta memasukkan *username* dan *password*. Kalau datanya benar, pengguna akan diarahkan ke menu sesuai dengan perannya, yaitu (Admin atau Sales). Tapi kalau salah, sistem akan memberikan tiga

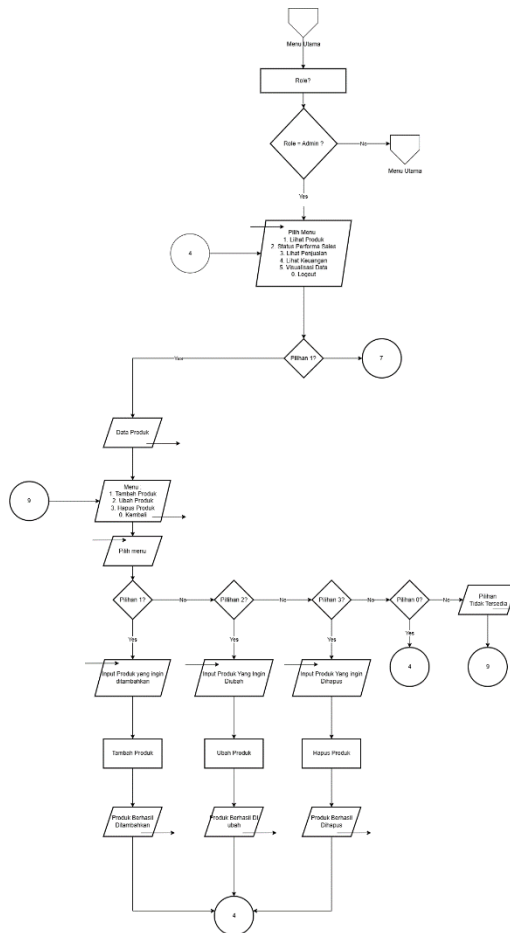
kesempatan untuk mencoba lagi. Kalau tetap salah setelah tiga kali, pengguna akan dikembalikan ke menu utama.



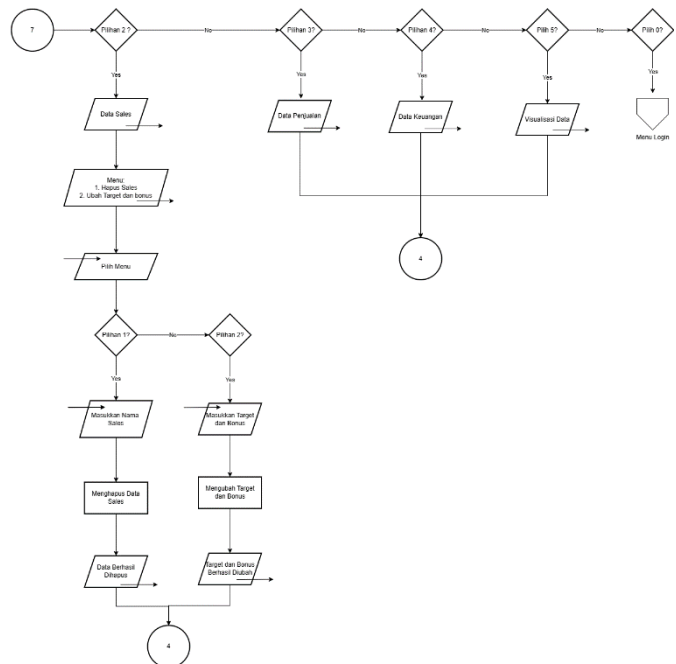
Kalau memilih *Register*, pengguna bisa membuat akun baru. Pertama, mereka diminta memasukkan *username*. Kalau *username* itu belum dipakai, pengguna lanjut memasukkan *password* dan memilih peran (Admin atau Sales). Kalau semuanya benar, akun baru berhasil dibuat, dan pengguna akan kembali ke menu utama dengan pesan "Registrasi Berhasil".

Kalau memilih Keluar, program akan menampilkan pesan "Terima Kasih Telah Menggunakan Program Kami" dan selesai. Tapi, kalau pengguna memilih angka yang salah di menu utama, sistem akan memberi tahu bahwa pilihannya tidak valid, lalu kembali ke menu utama.

Setelah setiap proses selesai, seperti *login*, *register*, atau salah pilih menu, program akan selalu kembali ke menu utama sampai pengguna memutuskan untuk keluar. Sistem ini dibuat sederhana agar mudah digunakan oleh siapa saja.



Gambar 2 3 Menu Admin



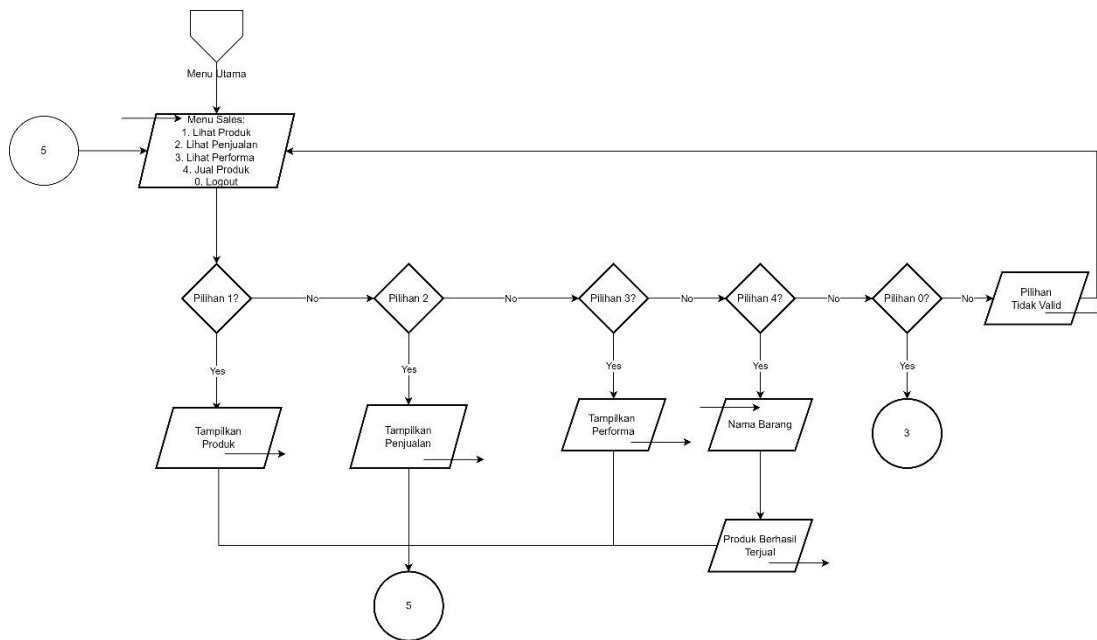
Gambar 2 2 Menu Admin

Ketika memilih menu admin, pengguna diarahkan ke menu utama admin. Admin memiliki lima pilihan: “1. Lihat Data Produk”, “2. Status Performa Sales”, “3. Lihat Penjualan”, “4. Lihat Keuangan”, dan “5. Visualisasi Data”. Pilihan 0 digunakan untuk kembali ke menu *login*.

Jika admin memilih “1”. Lihat Data Produk, mereka diarahkan ke sub-menu untuk mengelola produk. Admin bisa menambah produk, mengubah produk, atau menghapus produk. Admin harus memasukkan informasi spesifik, seperti nama atau detail produk, tergantung pada operasi yang dipilih. Setelah operasi selesai (misalnya, produk ditambahkan, diubah, atau dihapus), sistem akan mengonfirmasi bahwa tindakan berhasil, kemudian kembali ke menu produk.

Jika admin memilih “2”. Status Performa Sales, mereka memiliki dua opsi, yaitu: menghapus data sales atau mengubah *target* dan *bonus* sales. Untuk menghapus sales, admin harus memasukkan nama sales, lalu sistem akan menghapusnya setelah konfirmasi. Jika admin ingin mengubah *target* dan *bonus* sales, mereka dapat memasukkan target baru, dan sistem akan memperbarui data. Setelah selesai, admin akan kembali ke menu sales.

Pilihan “3”. Lihat Penjualan, memungkinkan admin melihat atau mengelola data penjualan. Pilihan “4”. Lihat Keuangan, digunakan untuk mengakses laporan keuangan, dan pilihan “5””. Visualisasi Data, digunakan untuk melihat data dalam bentuk grafik atau representasi visual lainnya. Setiap menu ini dirancang agar setelah selesai, admin kembali ke menu utama admin untuk melanjutkan operasi lainnya.



**Gambar 2. 1 Flowchart menu sales**

Setelah masuk ke menu utama untuk Sales, pengguna diberikan lima pilihan utama: “1. Lihat Produk”, “2. Lihat Penjualan”, “3. Lihat Performa”, “4. Jual Produk”, dan 0. Logout. Jika pengguna memilih angka selain yang tersedia, sistem akan memberikan peringatan bahwa pilihan tidak valid dan kembali ke menu utama.

Pilihan 1. Lihat Produk: Jika pengguna memilih opsi ini, sistem akan menampilkan daftar produk yang tersedia. Setelah itu, pengguna akan kembali ke menu utama untuk melanjutkan aktivitas lainnya. Pilihan 2. Lihat Penjualan: Opsi ini akan menampilkan data penjualan yang telah dicatat oleh sistem. Setelah selesai, pengguna akan kembali ke menu utama. Pilihan 3. Lihat Performa: Pada opsi ini, sistem akan menampilkan performa pengguna berdasarkan data penjualan mereka, seperti *target* penjualan atau jumlah produk yang terjual. Setelah informasi performa ditampilkan, pengguna diarahkan kembali ke menu utama. Pilihan 4. Jual Produk: Jika pengguna memilih untuk menjual produk, mereka akan diminta memasukkan nama barang yang ingin dijual. Setelah barang terjual, sistem akan memberikan konfirmasi bahwa produk



berhasil dijual, kemudian mengembalikan pengguna ke menu utama. Pilihan 0. *Logout*: Pilihan ini digunakan untuk keluar dari menu sales dan kembali ke halaman *login*.

Dari Flowchart yang dibuat, menunjukkan bahwa setiap aktivitas yang dilakukan oleh pengguna akan selalu kembali ke menu utama, sehingga memungkinkan pengguna untuk menjalankan tugas lainnya dengan mudah tanpa perlu keluar dari sistem. Alur ini memastikan fleksibilitas dalam pekerjaan sekaligus memberikan struktur yang jelas untuk pengguna dengan peran sales.

### **2.3 Konsep/Materi Praktikum yang dipakai**

Konsep yang kami gunakan dalam pembuatan laporan ini sesuai dengan persyaratan yang terdapat pada modul diantaranya:

1. Fungsi dasar: fungsi yang kami gunakan adalah fungsi *print*.  
Fungsi *print* pada program merupakan fungsi yang umum dipakai untuk menampilkan suatu keluaran pada layar peraga.
2. Fungsi percabangan : fungsi yang kami gunakan adalah fungsi *if*, *elif*, dan *else*.  
Fungsi tersebut dibuat dengan tujuan menentukan tindakan dan perintah sesuai kondisi yang digunakan.
3. Fungsi perulangan : fungsi yang kami gunakan adalah fungsi *while true* dan *for*.  
Fungsi tersebut digunakan untuk melakukan perintah secara berulang-ulang dengan jumlah yang ditentukan atau selama masih memenuhi kondisi yang ditentukan.
4. Fungsi list & tuple : fungsi yang kami gunakan adalah fungsi *list*.  
Fungsi tersebut digunakan untuk menyimpan berbagai tipe data sekaligus dan datanya bisa di edit.
5. Fungsi dictionary : fungsi yang kami gunakan adalah fungsi *dictionary*.

Fungsi ini digunakan untuk menyimpan kumpulan data/nilai yang dapat diakses menggunakan kata kunci dari data yang tersedia.

6. Fungsi CRUD : fungsi yang kami gunakan adalah fungsi CRUD (*create, read, update, delete*). Fungsi ini digunakan untuk mempermudah pengelolaan data dalam suatu sistem dimana kita bisa melihat, menambah, mengedit dan menghapus data yang ada.

7. Fungsi def : fungsi yang kami gunakan adalah fungsi def.

Fungsi ini digunakan untuk memecah program besar menjadi sub program yang lebih sederhana. Masing – masing fitur pada program dapat dibuat dalam satu fungsi, sehingga saat kita membutuhkan fitur tersebut kita hanya perlu memanggil fungsinya.

8. Fungsi error handling & file eksternal : fungsi yang kami gunakan dalam error handling adalah try-Except dan untuk file eksternalnya adalah csv.

Fungsi try-except digunakan untuk menangani error, sehingga program tidak langsung berhenti dan error ketika terjadi kesalahan saat eksekusi data. Dan penggunaan file eksternal csv adalah untuk menyimpan data.

## BAB III

### HASIL DAN PEMBAHASAN

#### 3.1 Tampilan Program

Tampilan menu awal

```
=====
=== SISTEM MANAJEMEN RETAIL ===
=====
1. Login
2. Register
0. Keluar
=====
Pilih menu: █
```

Gambar 3. 1

Tampilan Login

```
Pilih menu: 1
=== LOGIN ===
Username: admin
Password: █
```

Gambar 3. 2

### Tampilan Login Jika Salah

```
Pilih menu: 1

=== LOGIN ===
Username: sbdbff
Password:
Username atau password salah! Kesempatan login tersisa 2
Tekan enter untuk melanjutkan....
```

Gambar 3. 3

### Menu Admin

```
=====
Selamat datang, admin!
=====
|
|                                     === MENU ADMIN ===
|
|      1. Lihat Produk
|      2. Status Performa Sales
|      3. Lihat Penjualan
|      4. Lihat Keuangan
|      5. Visualisasi Data
|      0. Logout
|
=====
Pilih menu: 
```

Gambar 3. 4

### Lihat Produk

```
+-----+-----+-----+-----+
| ID | Nama produk |      Harga      | Stok |
+-----+-----+-----+-----+
| 2  |      Meja    | Rp 1,500,000.00 | 0    |
+-----+-----+-----+-----+

=== MENU PRODUK ===
1. Tambah Produk
2. Ubah Produk
3. Hapus Produk
0. Kembali
Pilih menu: 
```

Gambar 3. 5

### Tampilan Tambah Produk

```
=== TAMBAH PRODUK ===
ID Produk: 3
Nama Produk: kursi
Harga: 125000
Stok: 4
Produk berhasil ditambahkan! Tekan Enter untuk melanjutkan...
```

Gambar 3. 6

### Tampilan Ubah Produk

```
=== UBAH PRODUK ===  
Masukkan id produk yang akan diubah: 3  
Nama Produk baru: bantal  
Harga baru: 50000  
Stok baru: 30  
Produk berhasil diubah! Tekan Enter untuk melanjutkan...
```

Gambar 3. 7

### Tampilan Hapus Produk

```
=== HAPUS PRODUK ===  
Masukkan ID produk yang akan dihapus: 3  
Produk berhasil dihapus! Tekan Enter untuk melanjutkan...
```

Gambar 3. 8

### Tampilan Status Performa Sales

```
Performa Sales:
+-----+-----+-----+-----+
| Nama |          Target          | Bonus |
+-----+-----+-----+-----+
| ucup | Rp 100,000,000,000,000.00 | Rp 10.00 |
+-----+-----+-----+-----+

=== MENU PERFORMA SALES ===
1. Hapus Sales
2. Ubah Target dan Bonus Sales
0. Kembali
Pilih menu: █
```

Gambar 3. 9

### Tampilan Hapus Sales

```
=== MENU PERFORMA SALES ===
1. Hapus Sales
2. Ubah Target dan Bonus Sales
0. Kembali
Pilih menu: 1
Masukkan username sales yang akan dihapus: ucup
Sales berhasil dihapus! Tekan Enter untuk melanjutkan... █
```

Gambar 3. 10

### Tampilan Ubah Target dan Bonus Sales

```
=== MENU PERFORMA SALES ===
1. Hapus Sales
2. Ubah Target dan Bonus Sales
0. Kembali
Pilih menu: 2
Masukkan username sales: ucup
Target baru: 5000000
Bonus baru: 123000
Target dan bonus berhasil diubah! Tekan Enter untuk melanjutkan...
```

Gambar 3. 11

### Tampilan Lihat Penjualann

Tanggal	Id Produk	Jumlah	Total	Nama
2024-1-22 16:38:55	2	2	Rp 3,000,000.00	ucup
2024-1-22 16:40:27	2	1	Rp 1,500,000.00	ucup
2024-2-22 16:38:55	2	2	Rp 3,000,000.00	ucup
2024-3-22 16:40:27	2	1	Rp 1,500,000.00	ucup
2024-4-22 16:38:55	2	2	Rp 3,000,000.00	ucup
2024-11-22 16:40:27	2	1	Rp 1,500,000.00	ucup
2024-11-22 16:38:55	2	2	Rp 3,000,000.00	ucup
2024-12-22 16:40:27	2	1	Rp 1,500,000.00	ucup
2024-12-22 16:38:55	2	2	Rp 3,000,000.00	ucup
2024-12-22 16:40:27	2	1	Rp 1,500,000.00	ucup

Tekan Enter untuk melanjutkan...

Gambar 3. 12

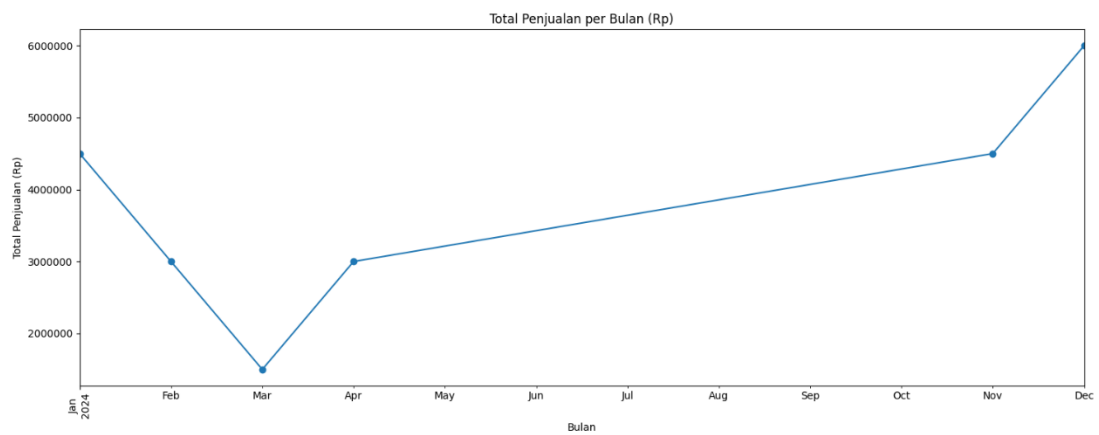


### Tampilan Lihat Keuangan

```
Total Pendapatan: Rp 22,500,000.00  
Tekan Enter untuk melanjutkan...
```

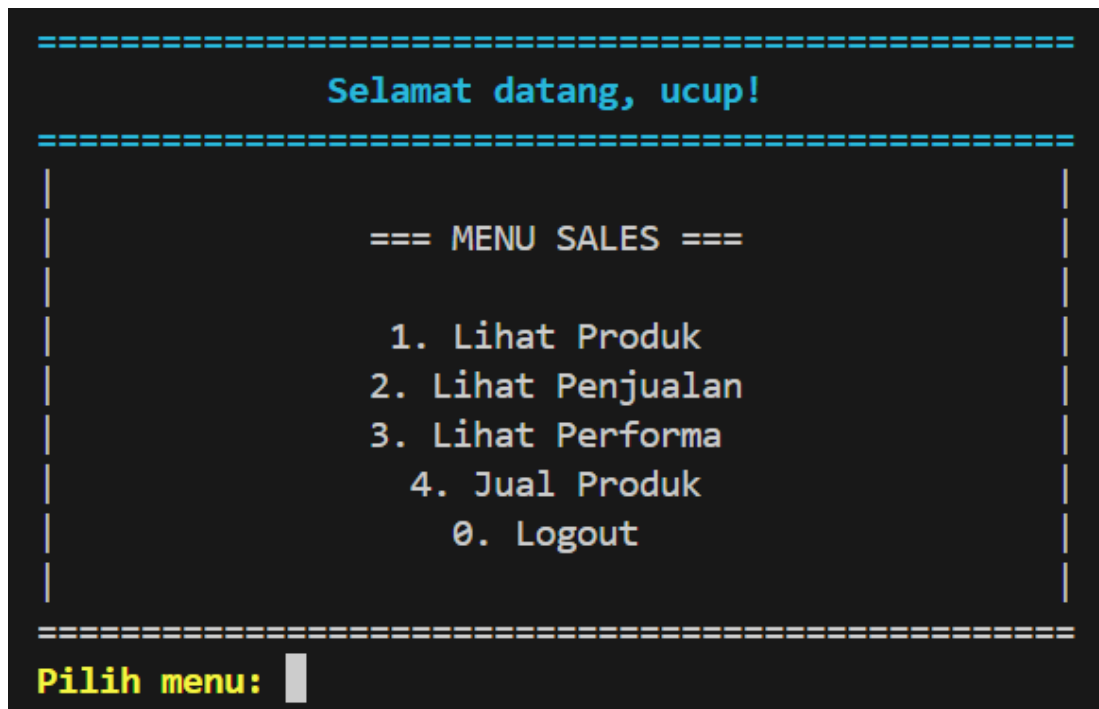
Gambar 3. 13

### Tampilan Visualisasi Data



Gambar 3. 14

### Tampilan Menu Sales



Gambar 3. 15

### Tampilan Menu Sales Lihat Produk

+---+-----+-----+-----+			
no	Nama produk	Price	Stock
+---+-----+-----+-----+			
2	Meja	Rp 1,500,000.00	0
+---+-----+-----+-----+			
Tekan enter untuk melanjutkan ...			

Gambar 3. 16

### Tampilan Menu Sales Lihat Penjualan

Tanggal	Id Produk	Jumlah	Harga	Nama
2024-1-22 16:38:55	2	2	Rp 3,000,000.00	ucup
2024-1-22 16:40:27	2	1	Rp 1,500,000.00	ucup
2024-2-22 16:38:55	2	2	Rp 3,000,000.00	ucup
2024-3-22 16:40:27	2	1	Rp 1,500,000.00	ucup
2024-4-22 16:38:55	2	2	Rp 3,000,000.00	ucup
2024-11-22 16:40:27	2	1	Rp 1,500,000.00	ucup
2024-11-22 16:38:55	2	2	Rp 3,000,000.00	ucup
2024-12-22 16:40:27	2	1	Rp 1,500,000.00	ucup
2024-12-22 16:38:55	2	2	Rp 3,000,000.00	ucup
2024-12-22 16:40:27	2	1	Rp 1,500,000.00	ucup

Tekan Enter untuk melanjutkan...

Gambar 3. 17

### Tampilan Menu Sales Lihat Performa

Performa Anda:  
Target: Rp 5,000,000.00  
Total Penjualan: Rp 22,500,000.00  
Progress: 450.00%  
Selamat! Anda mencapai target dan mendapatkan bonus Rp 123,000.00  
Tekan Enter untuk melanjutkan...

Gambar 3. 18

### Tampilan Menu Sales Jual Produk

```
=== JUAL PRODUK ===

Daftar Produk yang Tersedia:
+-----+-----+-----+-----+
| Id  | Nama Produk |      Harga      | Stok |
+-----+-----+-----+-----+
|  2  |      Meja   | Rp 1,500,000.00 |   5  |
+-----+-----+-----+-----+

Masukkan ID produk yang akan dijual: 2
Jumlah yang akan dijual: 3

Penjualan berhasil!
Total: Rp 4,500,000.00
Tekan Enter untuk melanjutkan...█
```

Gambar 3. 19

### Tampilan Menu Register

```
Pilih menu: 2

=== REGISTER ===
Username: (3-10 Karakter) nou
Password: (3-10 Karakter)

Pilih role:
1. Admin
2. Sales
Pilih role : 1
Masukkan kode autentikasi admin:
Registrasi berhasil!
Tekan Enter untuk melanjutkan
```

Gambar 3. 20

### Tampilan Menu Keluar



**Gambar 3. 21**

## 3.2 Source Code

```
1 pandas as pd
2 from getpass import getpass #import untuk input password
3 from colorama import Fore, Style, init
4 from admin import load
5 import re
6
7 def register():
8     try:
9         print("\n=== REGISTER ===")
10        username = input("Username: (3-10 Karakter) ")
11        if len(username) < 3: #jika username kurang dari 3 karakter akan terjadi nerror
12            raise ValueError(Fore.RED + "Username minimal 3 karakter")
13        if len(username) > 10: #jika username lebih dari 10 karakter akan terjadi error
14            raise ValueError(Fore.RED + "Username maksimal 10 karakter.")
15        if not username: #agar username tidak bisa kosong atau hanya input spasi
16            raise ValueError(Fore.RED + "Username tidak boleh kosong!" + Style.RESET_ALL)
17        if not re.match("[A-Za-z0-9]*$", username): #username hanya boleh berisi karakter a-z, A-Z, 0-9
18            raise ValueError(Fore.RED + "Username hanya boleh mengandung huruf dan angka tanpa karakter spesial!" + Style.RESET_ALL)
19
20        try:
21            userDF = pd.read_csv('user.csv') #membaca file user.csv
22        except FileNotFoundError: #jika file tidak ditemukan maka terjadi error (tidak menghentikan program)
23            print(Fore.RED + "Error: File 'user.csv' tidak ditemukan. Pastikan file tersedia." + Style.RESET_ALL)
24            return
25
26        if username in userDF['username'].values: #melakukan pengecekan apakah username sudah digunakan sebelumnya
27            input("Username sudah terdaftar! Tekan Enter untuk melanjutkan...")
28            return
29
30        password = getpass("Password: (3-10 Karakter)") #agar pada saat melakukan input tidak terlihat di terminal
31        if len(password) < 3:
32            raise ValueError(Fore.RED + "Password minimal 3 karakter")
33        if len(password) > 10:
34            raise ValueError(Fore.RED + "Password maksimal 10 karakter.")
35        if not password:
36            raise ValueError("Password tidak boleh kosong!")
37        print("\nPilih role:")
38        print("1. Admin")
39        print("2. Sales")
40        inputRole = input("Pilih role : ")
41
42        if inputRole == "1":
43            autentikasi = getpass("Masukkan kode autentikasi admin: ")
44            if autentikasi != "admin123": #jika kode selain ini maka kode dinyatakan tidak valid
45                input(Fore.RED + "Kode rahasia salah! Tekan Enter untuk kembali ke menu utama..." + Style.RESET_ALL)
46                return
47            role = "admin" #jika kode benar maka akan diberi role sebagai admin
48        elif inputRole == "2":
49            role = "Sales"
50        else:
51            input(Fore.RED + "Pilihan role tidak valid! Tekan Enter untuk kembali ke menu utama..." + Style.RESET_ALL)
52            return
53
54        if username and password and role:
55            userBaru = pd.DataFrame([username, password, role], columns=["username", "password", "role"]) #membuat variabel data baru
56            userDF = pd.concat([userDF, userBaru], ignore_index=True) #menggabungkan data baru
57            userDF.to_csv('user.csv', index=False) #simpan data pengguna baru ke file user.csv
58
59        if role == "Sales": #menambahkan target dan bonus secara otomatis jika memilih role sales
60            try:
61                salesDF = pd.read_csv('sales.csv')
62            except FileNotFoundError:
63                salesDF = pd.DataFrame(columns=["username", "target", "bonus"])
64
65            salesBaru = pd.DataFrame([username, 1000000, 100000], columns=["username", "target", "bonus"])
66            salesDF = pd.concat([salesDF, salesBaru], ignore_index=True)
67            salesDF.to_csv('sales.csv', index=False)
68
69        print(Fore.GREEN + "Registrasi berhasil!" + Style.RESET_ALL)
70
71    except ValueError as e:
72        print(Fore.RED + f"Input Error: {e}" + Style.RESET_ALL)
73    input("Tekan Enter untuk melanjutkan")
```

Gambar 3. 22

```

1 def login(): #fungsi login
2     kesempatan = 3
3     while kesempatan > 0:
4
5         try:
6             print("\n== LOGIN ==")
7             username = input("Username: ")
8             if not username:
9                 raise ValueError(Fore.RED + "Username tidak boleh kosong!" + Style.RESET_ALL)
10            if not re.match("^[A-Za-z0-9]*$", username):
11                raise ValueError(Fore.RED + "Username hanya boleh mengandung huruf dan angka tanpa karakter spesial!" + Style.RESET_ALL)
12            password = getpass("Password: ")
13            if not password:
14                raise ValueError(Fore.RED + "Password tidak boleh kosong" + Style.RESET_ALL)
15            userDF = pd.read_csv('user.csv')
16            user = userDF[(userDF['username'] == username) & (userDF['password'] == password)]
17
18            if not user.empty:
19                print(Fore.GREEN + "\nLogin berhasil! Tunggu sebentar" + Style.RESET_ALL)
20                load()
21                return {'username': username, 'role': user.iloc[0]['role']}
22            else:
23                kesempatan -= 1
24                print(Fore.RED + f"Username atau password salah! Kesempatan login tersisa {kesempatan}" + Style.RESET_ALL)
25
26                if kesempatan == 0:
27                    print("Kesempatan habis, anda akan keluar dari program")
28                    exit()
29
30            except ValueError as e:
31                print(Fore.RED + str(e) + Style.RESET_ALL)
32                input("Tekan enter untuk melanjutkan....")
33

```

**Gambar 3. 23**

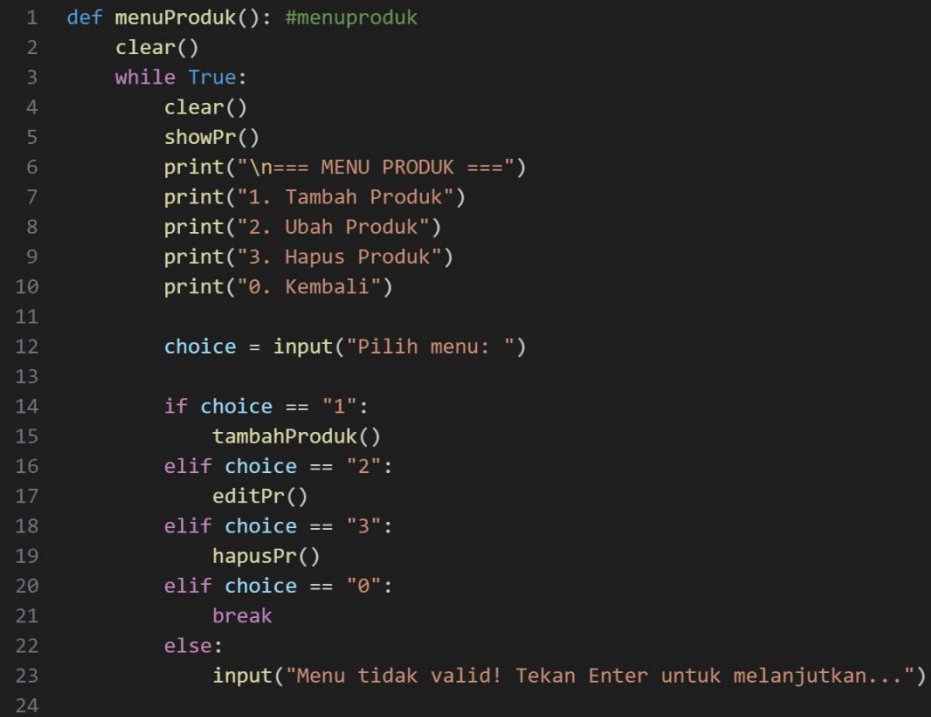


```

1  import os #buat clear
2  import pandas as pd #manipulasi data
3  import matplotlib.pyplot as plt #buat grafik
4  import time #buat jeda waktu
5  import sys #untuk mengubah runtime
6  from prettytable import PrettyTable #buat tabel
7  from datetime import datetime #buat tanggal
8  from colorama import Fore, Style, init #buat warna
9
10 def clear(): #buat clear terminal
11     os.system('cls || clear')
12
13 def load(panjangLoading=30, waktu=3.5): #loading bar
14     for i in range(panjangLoading + 1):
15         time.sleep(waktu / panjangLoading)
16         bar = "=" * i + "-" * (panjangLoading - i)
17         print(f"\rLoading: [{bar}] {i * 100 // panjangLoading}%", end="")
18
19
20 def showPr(): #show produk
21     clear()
22     produkDF = pd.read_csv('produk.csv')
23     if produkDF.empty:
24         print(Fore.RED + "Belum ada produk." + Style.RESET_ALL)
25         input("Tekan Enter untuk melanjutkan")
26     else:
27         table = PrettyTable()
28         table.field_names = ["ID", "Nama produk", "Harga", "Stok"] #isi headers
29
30         for _, row in produkDF.iterrows(): #baca tabel perbaris
31             rupiah = f"Rp {row['harga']:,.2f}" #mengubah pada kolom Harga agar menjadi format Rupiah
32             table.add_row([row['id'], row['nama'], rupiah, row['stok']])
33
34     print(table)

```

**Gambar 3. 24**



```
1 def menuProduk(): #menuproduk
2     clear()
3     while True:
4         clear()
5         showPr()
6         print("\n=== MENU PRODUK ===")
7         print("1. Tambah Produk")
8         print("2. Ubah Produk")
9         print("3. Hapus Produk")
10        print("0. Kembali")
11
12        choice = input("Pilih menu: ")
13
14        if choice == "1":
15            tambahProduk()
16        elif choice == "2":
17            editPr()
18        elif choice == "3":
19            hapusPr()
20        elif choice == "0":
21            break
22        else:
23            input("Menu tidak valid! Tekan Enter untuk melanjutkan...")
24
```

**Gambar 3. 25**

```

1 def tambahProduk():
2     try:
3         produkDF = pd.read_csv('produk.csv') #membaca file csv
4         print("\n=== TAMBAH PRODUK ===")
5         idProduk = input("ID Produk: ").strip() #menghapus spasi didepan maupun belakang
6         if not idProduk: #agar input tidak bisa kosong
7             raise ValueError(Fore.RED + "input tidak boleh kosong" + Style.RESET_ALL)
8         if not idProduk.isdigit(): #agar input hanya bisa angka
9             raise ValueError(Fore.RED + "Input harus berupa Angka" + Style.RESET_ALL)
10        if idProduk in produkDF['id'].astype(str).values: #melakukan pengecekan apakah id sudah ada di data sebelumnya
11            raise ValueError (Fore.RED + 'ID sudah tersedia di data' + Style.RESET_ALL)
12        nama = input("Nama Produk: ").strip()
13        if not nama:
14            raise ValueError(Fore.RED + 'input tidak boleh kosong' + Style.RESET_ALL)
15        hargaBaru = input("Harga: ").strip() #dibuat string terlebih dahulu agar lebih mudah pada error handling
16        if not hargaBaru:
17            raise ValueError(Fore.RED + "input tidak boleh kosong" + Style.RESET_ALL)
18        if not hargaBaru.isnumeric():
19            raise ValueError(Fore.RED + "Harga harus berupa Angka" + Style.RESET_ALL)
20        harga = float(hargaBaru) #mengubah jenis data menjadi float
21        stokBaru = input("Stok: ").strip()
22        if not stokBaru:
23            raise ValueError (Fore.RED + 'Input tidak boleh kosong' + Style.RESET_ALL)
24        if not stokBaru.isdigit():
25            raise ValueError (Fore.RED + 'Input harus berupa Angka' + Style.RESET_ALL)
26        stok = int(stokBaru) #mengubah data stok menjadi int agar tidak menjadi bilangan pecahan
27        produkBaru = pd.DataFrame({'id': [idProduk], 'nama': [nama], 'harga': [harga], 'stok': [stok]}) #menggunakan nilai nilai sebelumnya ke DF
28        produkDF = pd.concat([produkDF, produkBaru], ignore_index=True) #menggabungkan data
29        produkDF.to_csv('produk.csv', index=False) #melakukan push ke csv
30        input(Fore.GREEN + "Produk berhasil ditambahkan! Tekan Enter untuk melanjutkan..." + Style.RESET_ALL)
31    except ValueError as e:
32        print(e)
33    input("tekan enter untuk melanjutkan...")

```

Gambar 3. 26

```

1 def editPr():
2     while True:
3         print("\n== UBAH PRODUK ==")
4         idInput = input("Masukkan id produk yang akan diubah: ").strip()
5         if idInput.isalpha():
6             print(fore.RED + "Input harus berupa Angka" + style.RESET_ALL)
7             input("Tekan Enter untuk melanjutkan")
8             return
9
10        if idInput:
11            try:
12                idProduk = int(idInput)
13                produkDF = pd.read_csv('produk.csv')
14
15                if idProduk in produkDF['id'].values:
16
17                    Produk = produkDF.loc[produkDF['id'] == idProduk].iloc[0] #mencari id produk sesuai dengan input yang dilakukan
18
19
20                    nama = input("Nama Produk baru: ").strip()
21                    harga = input("Harga baru: ").strip()
22                    if harga.isalpha():
23                        raise ValueError ("Input harus berupa Angka")
24                    stok = input("Stok baru: ").strip()
25                    if stok.isalpha():
26                        raise ValueError ("Input harus berupa Angka")
27
28
29                    namaBaru = nama if nama else Produk['nama'] #menggunakan nilai lama jika input kosong
30                    hargaBaru = float(harga) if harga.isdigit() else Produk['harga'] #menggunakan nilai lama jika input kosong
31                    stokBaru = int(stok) if stok.isdigit() else Produk['stok'] #menggunakan nilai lama jika input kosong
32
33
34                    produkDF.loc[produkDF['id'] == idProduk, ['nama', 'harga', 'stok']] = [namaBaru, hargaBaru, stokBaru] #mengubah nilai data lama menjadi data baru
35                    produkDF.to_csv('produk.csv', index=False) #push ke csv
36
37                    input(fore.GREEN + "Produk berhasil diubah! Tekan Enter untuk melanjutkan..." + style.RESET_ALL)
38                    break
39            except:
40                input(fore.RED + "Produk tidak ditemukan! Tekan Enter untuk melanjutkan..." + style.RESET_ALL)
41                break
42        except ValueError as e:
43            print(fore.RED + f"Terjadi Kesalahan (e)" + style.RESET_ALL)
44            input("Tekan Enter untuk melanjutkan")
45            return
46        else:
47            print(fore.RED + "Input tidak boleh kosong." + style.RESET_ALL)
48            input("Tekan Enter untuk mencoba lagi...")
49            return

```

**Gambar 3. 27**

```

1 def hapusPr():
2     try:
3         print("\n== HAPUS PRODUK ==")
4         idProdukBaru = input("Masukkan ID produk yang akan dihapus: ").strip()
5         if not idProdukBaru:
6             raise ValueError (Fore.RED + 'Input tidak boleh kosong ' + Style.RESET_ALL) #error handling jika melakukan input kosong
7         if idProdukBaru.isalpha():
8             raise ValueError (Fore.RED + 'Input harus berupa angka' + Style.RESET_ALL)
9
10        idProduk = int(idProdukBaru)
11        produkDF = pd.read_csv('produk.csv')
12        if idProduk in produkDF['id'].values:
13            produkDF = produkDF[produkDF['id'] != idProduk] #membuat DF baru dimana hanya id yang tidak sama yg disimpan
14            produkDF.to_csv('produk.csv', index=False)
15            input("Produk berhasil dihapus! Tekan Enter untuk melanjutkan...")
16        else:
17            input(Fore.RED + "Produk tidak ditemukan! Tekan Enter untuk melanjutkan..." + Style.RESET_ALL)
18    except ValueError as e:
19        print(Fore.RED + f"Terjadi Kesalahan {e}" + Style.RESET_ALL)
20        input("Tekan Enter untuk melanjutkan")

```

**Gambar 3. 28**

```

1  def performaSales():
2      clear()
3      while True:
4          clear()
5          salesDF = pd.read_csv('sales.csv')
6          if salesDF.empty:
7              print(Fore.RED + "Tidak ada data!" + Style.RESET_ALL)
8          else:
9              print("\nPerforma Sales:")
10             tabelSales = PrettyTable()
11             tabelSales.field_names = ["Nama", "Target", "Bonus"]
12
13             for _, row in salesDF.iterrows():
14                 rupiahSales = f"Rp {row['target']:,.2f}" #membuat format rupiah
15                 bonusSales = f"Rp {row['bonus']:,.2f}" #membuat format rupiah
16                 tabelSales.add_row([row['username'], rupiahSales, bonusSales])
17             print(tabelSales)
18
19
20             print("\n=== MENU PERFORMA SALES ===")
21             print("1. Hapus Sales")
22             print("2. Ubah Target dan Bonus Sales")
23             print("0. Kembali")
24
25             choice = input("Pilih menu: ")
26
27             if choice == "1":
28                 hapusSales()
29             elif choice == "2":
30                 editTarget()
31             elif choice == "0":
32                 break
33             else:
34                 input(Fore.RED + "Menu tidak valid! Tekan Enter untuk melanjutkan..." + Style.RESET_ALL)

```

**Gambar 3. 29**

```

1 def hapusSales():
2     username = input("Masukkan username sales yang akan dihapus: ")
3
4     salesDF = pd.read_csv('sales.csv')
5     userDF = pd.read_csv('user.csv')
6
7     if username in salesDF['username'].values:
8         salesDF = salesDF[salesDF['username'] != username] #hanya menyimpan data yang tidak sama dengan input
9         userDF = userDF[userDF['username'] != username] #hanya menyimpan data yang tidak sama dengan input
10
11         salesDF.to_csv('sales.csv', index=False)
12         userDF.to_csv('user.csv', index=False)
13         input(Fore.RED + "Sales berhasil dihapus! Tekan Enter untuk melanjutkan..." + Style.RESET_ALL)
14     else:
15         input(Fore.RED + "Sales tidak ditemukan! Tekan Enter untuk melanjutkan..." + Style.RESET_ALL)

```

**Gambar 3. 30**

```

1 def editTarget():
2     try:
3         username = input("Masukkan username sales: ").strip()
4
5         salesDF = pd.read_csv('sales.csv')
6         if username in salesDF['username'].values:
7             targetBaru = input("Target baru: ") #menggunakan data str agar lebih mudah pada saat error handling
8             if not targetBaru:
9                 raise ValueError (Fore.RED + 'Input tidak boleh kosong' + Style.RESET_ALL)
10            if not targetBaru.isnumeric(): #jika input bukan angka maka akan terjadi error
11                raise ValueError (Fore.RED + 'Input harus berupa Angka' + Style.RESET_ALL)
12            target = float(targetBaru) #mengubah jenis data
13
14            bonusBaru = input("Bonus baru: ")
15            if not bonusBaru:
16                raise ValueError (Fore.RED + 'Input tidak boleh kosong' + Style.RESET_ALL)
17            if not bonusBaru.isnumeric():
18                raise ValueError (Fore.RED + 'Input harus berupa angka' + Style.RESET_ALL)
19            bonus = float(bonusBaru)
20
21            salesDF.loc[salesDF['username'] == username, ['target', 'bonus']] = [target, bonus]
22            salesDF.to_csv('sales.csv', index=False)
23            input("Target dan bonus berhasil diubah! Tekan Enter untuk melanjutkan...")
24        except ValueError as e:
25            print(Fore.RED + "Terjadi kesalahan {e}" + Style.RESET_ALL)
26            input("Tekan Enter untuk melanjutkan")

```

**Gambar 3. 31**

```

1 def dataSales():
2     clear()
3     transaksiDF = pd.read_csv('history.csv')
4     if transaksiDF.empty:
5         print("Belum ada penjualan.")
6     else:
7         tabelPenjualan = PrettyTable()
8         tabelPenjualan.field_names = ["Tanggal", "Id Produk", "Jumlah", "Total", "Nama"]
9         for _, row in transaksiDF.iterrows():
10             rupiah = f"Rp {row['total']:,.2f}" #mengubah format menjadi rupiah
11             idProduk = int(row['id_produk']) #mengubah format menjadi int
12             tabelPenjualan.add_row([row['tanggal'], idProduk, row['jumlah'], rupiah, row['nama_sales']])
13         print(tabelPenjualan)
14         input("\nTekan Enter untuk melanjutkan...")

```

**Gambar 3. 32**

```

1 def dataKeuangan():
2     clear()
3     transaksiDF = pd.read_csv('history.csv')
4     if transaksiDF.empty:
5         print("Belum ada data keuangan.")
6     else:
7         total_income = transaksiDF['total'].sum() #menjumlahkan total pendapatan
8         print(f"\nTotal Pendapatan: Rp {total_income:,.2f}")
9         input("\nTekan Enter untuk melanjutkan...")

```

**Gambar 3. 33**



```

1 def visualisasi():
2
3     transaksiIDF = pd.read_csv('history.csv')
4
5
6     if transaksiIDF.empty:
7         print("Belum ada data untuk divisualisasikan.")
8         input("Tekan Enter Untuk Melanjutkan")
9         return
10
11     transaksiIDF['tanggal'] = pd.to_datetime(transaksiIDF['tanggal'], dayfirst=True) #mengubah menjadi dd/mm/yy
12
13     penjualanBulanan = transaksiIDF.groupby(transaksiIDF['tanggal'].dt.to_period('M'))['total'].sum() #mengelompokkan data berdasarkan bulan dan menjumlahkan total setiap bulan
14
15     plt.figure(figsize=(15, 6)) #membuat gambar dengan ukuran 15x6
16     penjualanBulanan.plot(kind='line', marker='o') #menambah titik pada setiap data point
17     plt.title('Total Penjualan per Bulan (Rp)') #judul
18     plt.xlabel('Bulan') #sumbu x
19     plt.ylabel('Total Penjualan (Rp)') #sumbu y
20     plt.xticks(rotation=90) #memutar 90 derajat agar lebih mudah dilihat
21
22
23     plt.ticklabel_format(style='plain', axis='y') #menampilkan angka dalam format normal
24
25     plt.tight_layout() #mengatur spacing secara otomatis
26     plt.show(block=True) #membuat program akan menunggu sampai grafik di tutup oleh pengguna
27
28 def midPrint(text, width=50):
29     spaces = (width - len(text)) // 2 #menghitung jumlah spasi yang dibutuhkan di setiap sisi
30     extra_space = (width - len(text)) % 2 #menghitung spasi jika panjangnya ganjil
31     return " " * spaces + text + " " * (spaces + extra_space) #mengembalikan dan menambahkan
32
33 def adminMenu(username):
34     clear()
35     width = 50 # Lebar bingkai
36     while True:
37         clear()
38         # Membuat tampilan header
39         print(fore.CYAN + style.BRIGHT + "-" * width)
40         print(midPrint(f'Selamat datang, {username}!', width))
41         print("-" * width + style.RESET_ALL)
42
43         # Menampilkan menu di tengah bingkai
44         print("|" + " " * (width - 2) + "|")
45         print("|" + midPrint("=== MENU ADMIN ===", width - 2) + "|")
46         print("|" + " " * (width - 2) + "|")
47         print("|" + midPrint("1. Lihat Produk", width - 2) + "|")
48         print("|" + midPrint("2. Status Performa Sales", width - 2) + "|")
49         print("|" + midPrint("3. Lihat Penjualan", width - 2) + "|")
50         print("|" + midPrint("4. Lihat Keuangan", width - 2) + "|")
51         print("|" + midPrint("5. Visualisasi Data", width - 2) + "|")
52         print("|" + midPrint("0. Logout", width - 2) + "|")
53         print("|" + " " * (width - 2) + "|")
54         print("-" * width)
55
56         # Input pilihan
57         choice = input(fore.YELLOW + style.BRIGHT + "Pilih menu: " + style.RESET_ALL)
58
59         # Menu navigasi
60         if choice == "1":
61             menuProduk()
62         elif choice == "2":
63             performaSales()
64         elif choice == "3":
65             dataSales()
66         elif choice == "4":
67             dataKeuangan()
68         elif choice == "5":
69             visualisasi()
70         elif choice == "0":
71             break
72         else:
73             input("Menu tidak valid! Tekan Enter untuk melanjutkan...")
74

```

Gambar 3. 34

```
1  import os
2  import pandas as pd
3  from colorama import Fore, Style, init
4  from prettytable import PrettyTable
5  from datetime import datetime
6  from admin import midPrint, clear
7
8  def produk():
9      clear()
10     produkDF = pd.read_csv('produk.csv')
11     if produkDF.empty:
12         print("Belum ada produk.")
13     else:
14         table = PrettyTable()
15         table.field_names = ["no", "Nama produk", "Price", "Stock"]
16
17         for _, row in produkDF.iterrows():
18             rupiah = f"Rp {row['harga']:,.2f}"
19             table.add_row([row['id'], row['nama'], rupiah, row['stok']])
20
21         print(table)
22     input("Tekan enter untuk melanjutkan ...")
```

Gambar 3. 35

```

1 def dataSales(username):
2     clear()
3     transaksiDF = pd.read_csv('history.csv') #membaca file csv
4     transaksiUser = transaksiDF[transaksiDF['nama_sales'] == username] #melakukan pengecekan username yang login
5     if transaksiUser.empty:
6         print("Anda belum memiliki penjualan.")
7     else:
8         showSales = PrettyTable()
9         showSales.field_names = ["Tanggal", "Id Produk", "Jumlah", "Harga", "Nama"]
10
11         for _, row in transaksiDF.iterrows(): #membaca file perbaris
12             rupiah = f"Rp {row['total']:,.2f}"
13             showSales.add_row([row['tanggal'], row['id_produk'], row['jumlah'], rupiah, row['nama_sales']])
14         print(showSales)
15
16     input("\nTekan Enter untuk melanjutkan...")
17

```

Gambar 3. 36

```

1 def performa(username):
2     clear()
3     salesDF = pd.read_csv('sales.csv') #membaca file csv ke df
4     transaksiDF = pd.read_csv('history.csv') #membaca file csv ke df
5
6     if username not in salesDF['username'].values:
7         print("Anda belum memiliki penjualan.")
8         input ('tekan enter untuk melanjutkan')
9         return
10
11     sales = salesDF[salesDF['username'] == username]
12     target = sales['target'].values[0]
13     bonus = sales['bonus'].values[0]
14
15     totalSales = transaksiDF[transaksiDF['nama_sales'] == username]['total'].sum() #melakukan penjumlahan
16
17     print(f"\nPerforma Anda:")
18     print(f"Target: Rp {target:,.2f}")
19     print(f"Total Penjualan: Rp {totalSales:,.2f}")
20     print(f"Progress: {(totalSales/target)*100:.2f}%")
21     if totalSales >= target:
22         print(f"Selamat! Anda mencapai target dan mendapatkan bonus Rp {bonus:,.2f}")
23     else:
24         sisa = target - totalSales
25         print(f"Sisa target yang harus dicapai: Rp {sisa:,.2f}")
26     input("\nTekan Enter untuk melanjutkan...")

```

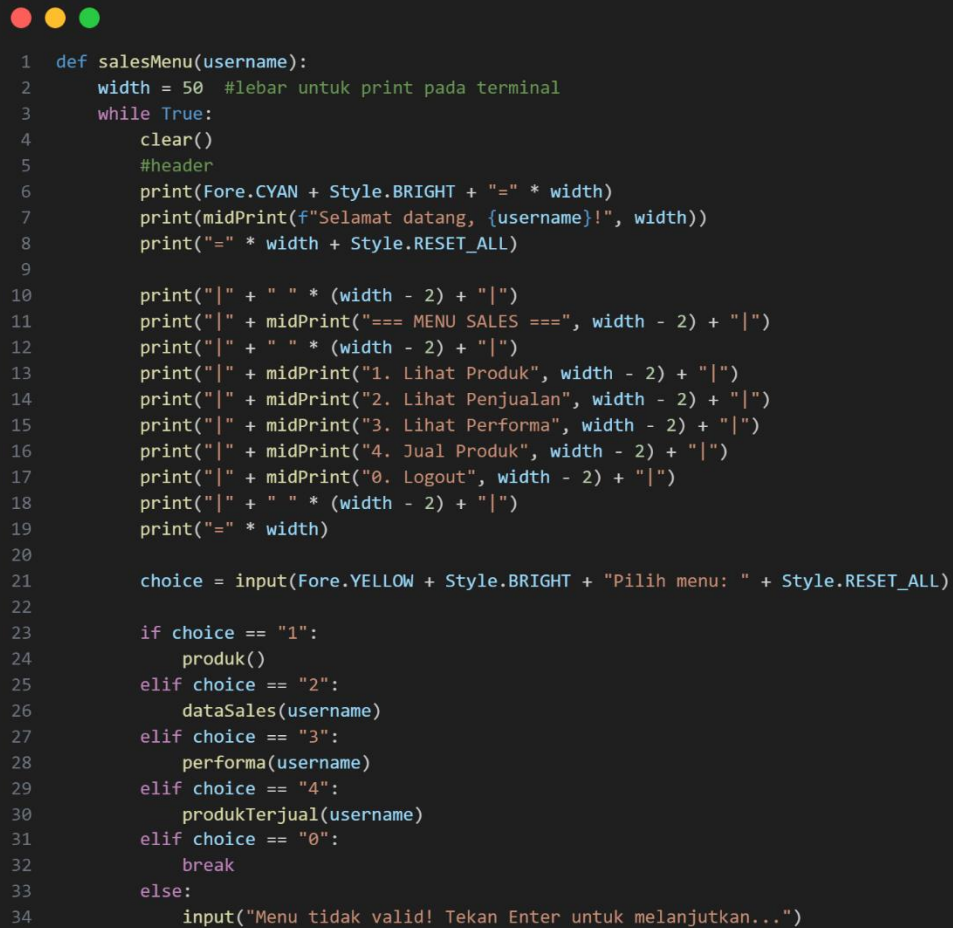
Gambar 3. 37

```

1  def produkTerjual(username):
2      clear()
3      try:
4          print("\n=== JUAL PRODUK ===")
5          produk_df = pd.read_csv('produk.csv')
6
7          if produk_df.empty:
8              input("Tidak ada produk yang tersedia. Tekan Enter untuk melanjutkan...")
9              return
10
11         print("\nDaftar Produk yang Tersedia:")
12         tabelJual = PrettyTable()
13         tabelJual.field_names = ["Id ", "Nama Produk", "Harga", "Stok"]
14
15         for _, row in produk_df.iterrows():
16             rupiahHarga = f"Rp {row['harga']:, .2f}"
17             tabelJual.add_row([row['id'], row['nama'], rupiahHarga, row['stok']])
18         print(tabelJual)
19
20         id_produkBaru = input("\nMasukkan ID produk yang akan dijual: ").strip() #menghapus spasi didepan maupun dibelakang
21         if not id_produkBaru:
22             raise ValueError("Input tidak boleh kosong")
23         if id_produkBaru.isalpha():
24             raise ValueError("Input harus berupa angka")
25
26         id_produk = int(id_produkBaru)
27
28         if int(id_produk) in produk_df['id'].values:
29             product = produk_df[produk_df['id'] == int(id_produk)]
30             jumlah = int(input("Jumlah yang akan dijual: "))
31             if not jumlah:
32                 raise ValueError("Input tidak boleh kosong")
33             if jumlah <= 0:
34                 print("Stok telah habis")
35
36             if jumlah <= product['stok'].iloc[0]:
37                 total = jumlah * product['harga'].iloc[0] #iloc berfungsi untuk membaca int sesuai namanya (integer location)
38                 tanggal = datetime.now().strftime('%Y-%m-%d %H:%M:%S') #format tanggal
39
40                 produk_df.loc[produk_df['id'] == int(id_produk), 'stok'] -= jumlah #melakukan pengurangan stok
41                 produk_df.to_csv('produk.csv', index=False)
42
43                 transaksiDF = pd.read_csv('history.csv')
44                 transaksiBaru = pd.DataFrame({
45                     'tanggal': [tanggal],
46                     'id_produk': [id_produk],
47                     'jumlah': [jumlah],
48                     'total': [total],
49                     'nama_sales': [username]
50                 })
51                 transaksiDF = pd.concat([transaksiDF, transaksiBaru], ignore_index=True) #menggabungkan data baru
52                 transaksiDF.to_csv('history.csv', index=False)
53
54                 print(f"\nPenjualan berhasil!")
55                 print(f"Total: Rp {total:, .2f}")
56                 input("Tekan Enter untuk melanjutkan...")
57             else:
58                 input("Stok tidak mencukupi! Tekan Enter untuk melanjutkan...")
59         else:
60             input("Produk tidak ditemukan! Tekan Enter untuk melanjutkan...")
61     except ValueError as e:
62         print(e)
63         input("Tekan Enter untuk melanjutkan")

```

**Gambar 3. 38**

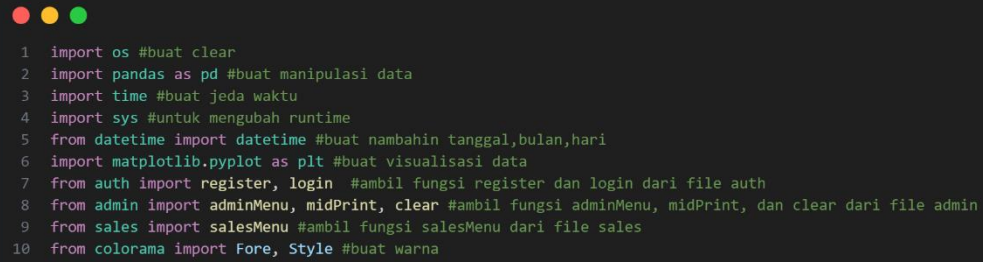


```

1  def salesMenu(username):
2      width = 50 #lebar untuk print pada terminal
3      while True:
4          clear()
5          #header
6          print(Fore.CYAN + Style.BRIGHT + "=" * width)
7          print(midPrint(f"Selamat datang, {username}!", width))
8          print("=" * width + Style.RESET_ALL)
9
10         print("|" + " " * (width - 2) + "|")
11         print("|" + midPrint("=== MENU SALES ===", width - 2) + "|")
12         print("|" + " " * (width - 2) + "|")
13         print("|" + midPrint("1. Lihat Produk", width - 2) + "|")
14         print("|" + midPrint("2. Lihat Penjualan", width - 2) + "|")
15         print("|" + midPrint("3. Lihat Performa", width - 2) + "|")
16         print("|" + midPrint("4. Jual Produk", width - 2) + "|")
17         print("|" + midPrint("0. Logout", width - 2) + "|")
18         print("|" + " " * (width - 2) + "|")
19         print("=" * width)
20
21         choice = input(Fore.YELLOW + Style.BRIGHT + "Pilih menu: " + Style.RESET_ALL)
22
23         if choice == "1":
24             produk()
25         elif choice == "2":
26             dataSales(username)
27         elif choice == "3":
28             performa(username)
29         elif choice == "4":
30             produkTerjual(username)
31         elif choice == "0":
32             break
33         else:
34             input("Menu tidak valid! Tekan Enter untuk melanjutkan...")

```

Gambar 3. 39

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains ten lines of Python code, each with a line number from 1 to 10 on the left. The code imports various modules and functions for a project, including os, pandas, time, sys, datetime, matplotlib, and custom modules like auth, admin, sales, and colorama.

```
1 import os #buat clear
2 import pandas as pd #buat manipulasi data
3 import time #buat jeda waktu
4 import sys #untuk mengubah runtime
5 from datetime import datetime #buat nambahin tanggal,bulan,hari
6 import matplotlib.pyplot as plt #buat visualisasi data
7 from auth import register, login #ambil fungsi register dan login dari file auth
8 from admin import adminMenu, midPrint, clear #ambil fungsi adminMenu, midPrint, dan clear dari file admin
9 from sales import salesMenu #ambil fungsi salesMenu dari file sales
10 from colorama import Fore, Style #buat warna
```

**Gambar 3. 40**

```

1  def mainMenu():
2      width = 50 #lebar
3      while True:
4          clear()
5          # Header
6          print(Fore.CYAN + Style.BRIGHT + "=" * width)
7          print(midPrint("=== SISTEM MANAJEMEN RETAIL ===", width))
8          print("=" * width + Style.RESET_ALL)
9          print("|" + " " * (width - 2) + "|")
10         print("|" + midPrint("1. Login", width - 2) + "|")
11         print("|" + midPrint("2. Register", width - 2) + "|")
12         print("|" + midPrint("0. Keluar", width - 2) + "|")
13         print("|" + " " * (width - 2) + "|")
14         print("=" * width)
15
16         choice = input(Fore.YELLOW + Style.BRIGHT + "Pilih menu: " + Style.RESET_ALL)
17
18         if choice == "1":
19             user = login()
20             if user is not None:
21                 if user['role'] == 'admin':
22                     adminMenu(user['username'])
23                 else:
24                     salesMenu(user['username'])
25         elif choice == "2":
26             register()
27         elif choice == "0":
28             kalimat = ""Terimakasih telah menggunakan program!\n
29
30             .....
31
32             .....
33             .....
34             .....
35             .....
36             .....
37             .....
38             .....
39             .....
40             .....
41             .. . \nHave a nice day!""
42             for _ in kalimat:
43                 sys.stdout.write(_) #menulis perhuruf
44                 time.sleep(0.005) #jeda penulisan perhuruf
45             break
46         else:
47             input("Menu tidak valid! Tekan Enter untuk melanjutkan...")
48
49 if __name__ == "__main__":
50     mainMenu()
51

```

Gambar 3. 41

## **BAB IV**

### **PENUTUP**

#### **4.1 Kesimpulan**

Program manajemen keuangan ini dirancang untuk perusahaan ritel, dengan fitur utama seperti manajemen produk, penjualan, performa sales, laporan keuangan, dan visualisasi data. Terdapat dua jenis pengguna, admin dan sales, dengan fitur yang disesuaikan sesuai peran. Admin dapat mengelola produk, memantau performa sales, dan menganalisis keuangan, sementara sales dapat mencatat penjualan dan memantau pencapaian *target*. Program ini disusun secara modular dalam beberapa file seperti “admin.py” dan “sales.py”, menggunakan file CSV untuk menyimpan data. Tujuannya adalah mempermudah pengelolaan keuangan, menyediakan laporan *real-time*, dan mendukung analisis bisnis untuk efisiensi dan pengambilan keputusan

#### **4.2 Saran**



Berdasarkan rumusan masalah dan tujuan program yang telah dijelaskan, terdapat beberapa saran yang dapat kami berikan untuk meningkatkan efektivitas dan efisiensi pengelolaan data jamaah haji di masa depan, yaitu:



1. Notifikasi Otomatis fitur notifikasi otomatis bertujuan untuk memberikan peringatan kepada admin jika stok suatu produk hampir habis.
2. Autentikasi lanjutan dengan enkripsi *password* adalah langkah penting untuk meningkatkan keamanan data pengguna.
3. Mengembangkan dashboard berbasis web akan membuat program lebih fleksibel dan mudah diakses kapan saja dari perangkat apa pun, seperti laptop atau smartphone.



## **DAFTAR PUSTAKA**

## LAMPIRAN

Tanggal Konsultasi : 31 OKTOBER 2024	
Uraian / Pembahasan : Flowchart , konsep catatan : beri balasan untuk register	
Asisten Lab  Nama: Adi Muhammad Syifai	Ketua Kelompok  Nama: M. Nabil Rahmatullah

Tanggal Konsultasi : 13 NOVEMBER 2024	
Uraian / Pembahasan : Uji program dan bug catatan : perbaikan pada register	
Asisten Lab  Nama: Adi Muhammad Syifai	Ketua Kelompok  Nama: M. Nabil Rahmatullah

Gambar 4. 1