LAPORAN PROYEK AKHIR
DASAR DASAR PEMROGRAMAN
SISTEM MANAJEMEN PAKET INTERNET & PULSA

Disusun Oleh:

KELOMPOK C9

SITI NURSINTA	**2509116087**
M. RIZKY ALFA REZA BASYAH	**2509116086**
ADELIA GITHA NAVEEZHA H.	**2509116110**

Asisten Laboratorium:

TAUFIK RAMADHANI	**DWI PEBRIYANTO PRADANA**
2509116001	**2509116012**

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS TEKNIK

UNIVERSITAS MULAWARMAN

2025

KATA PENGANTAR

Dengan menyebut nama Allah Yang Maha Pengasih dan Maha Penyayang, segala puji dan syukur kami panjatkan dan haturkan pada ke hadirat-Nya. Shalawat serta salam semoga selalu tercurahkan kepada junjungan kita, Nabi Muhammad SAW, keluarga, sahabat, dan para pengikut setianya. Kami dengan rendah hati dan bersungguh-sungguh menyusun laporan proyek akhir.

ini sebagai salah satu syarat kelulusan ujian akhir semester di Universitas Mulawarman, Kalimantan Timur. Laporan ini dibuat dalam rangka mengeksplorasi dan mendokumentasikan proyek pemrograman kami yang berfokus pada sistem Manajemen Paket Internet & Pulsa. Penyusunan laporan ini tidak mungkin terwujud tanpa bimbingan, dukungan, serta dorongan dari berbagai pihak yang tentunya ikut membantu perkembangan laporan ini.

Kami tentunya ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bang Dwi Pebriyanto Pradana selaku Asisten Laboratorium Konsul yang telah memberikan arahan, saran, serta pengawasan sepanjang perjalanan kami di proses pengerjaan PA.
2. Para abang dan mba selaku Asisten Laboratorium Sistem Informasi 2025
3. Seluruh teman mahasiswa semua yang sudah berkenan memberikan dukungan dan bantuan.

Kami menyadari bahwa laporan ini tidak sempurna, untuk itu segala kritik dan saran yang membangun sangat kami hargai dan untuk perbaikan di masa mendatang. Semoga laporan tugas akhir ini dapat memberikan pemahaman yang baik tentang proyek kami dan bermanfaat bagi semua pihak yang membacanya.

Oktober, 2025

Penyusun

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Deskripsi Masalah

Di era digital seperti sekarang, kebutuhan masyarakat terhadap layanan komunikasi dan internet semakin tinggi. Hampir semua kegiatan sehari-hari seperti belajar, bekerja, hingga hiburan membutuhkan koneksi internet dan pulsa sebagai sarana utama komunikasi (Wahyudi & Rose Handayani, 2025).

Pulsa merupakan nilai atau saldo yang digunakan pelanggan untuk mengakses berbagai layanan operator seluler, seperti melakukan panggilan, mengirim pesan, atau membeli paket internet. Sedangkan paket internet adalah layanan data yang disediakan oleh operator agar pengguna bisa mengakses jaringan internet sesuai kuota, masa aktif, dan kecepatan yang telah ditentukan.

Namun, dalam pengelolaan usaha penjualan pulsa dan paket internet, masih banyak ditemukan kendala. Misalnya, pencatatan transaksi yang masih manual, kesulitan memantau stok produk, serta kesalahan dalam laporan penjualan yang sering terjadi akibat kurang terorganisirnya sistem manajemen. Hal ini dapat menghambat kinerja usaha dan menurunkan efisiensi pelayanan kepada pelanggan.

Untuk mengatasi masalah tersebut, diperlukan Sistem Manajemen Paket Internet dan Pulsa yang mampu membantu pengelola dalam mencatat transaksi, mengelola data produk, serta menyimpan informasi pelanggan dengan lebih cepat, akurat, dan terstruktur. Dengan adanya sistem ini, kegiatan operasional konter atau agen pulsa dapat berjalan lebih efektif dan efisien.

Sistem ini dibuat menggunakan bahasa pemrograman Python dan dirancang memiliki dua peran pengguna, yaitu Admin dan User. Admin memiliki akses penuh untuk mengatur data produk dan transaksi, sementara user dapat melihat daftar produk dan melakukan pembelian. Melalui sistem ini, diharapkan pengelolaan usaha pulsa dan paket internet menjadi lebih modern, teratur, dan mudah digunakan.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang dapat diambil adalah sebagai berikut:

1. Bagaimana merancang Sistem Manajemen Paket Internet dan Pulsa yang efisien dan mudah digunakan?
2. Bagaimana sistem dapat dijalankan oleh Admin dengan akses penuh terhadap data produk dan transaksi?
3. Bagaimana sistem dapat dijalankan oleh User agar dapat melihat dan membeli produk dengan mudah?
4. Bagaimana sistem dapat membantu pencatatan dan pelaporan transaksi secara otomatis dan akurat?

5. Bagaimana penerapan sistem CRUD (Create, Read, Update, Delete) dalam pengelolaan data produk dan pengguna?

1.3 Batasan Masalah

Untuk menjaga agar pembahasan tetap fokus, maka batasan masalah pada sistem ini adalah:

1. Sistem hanya berfokus pada pengelolaan data produk berupa paket internet dan pulsa.
2. Sistem memiliki dua role utama, yaitu Admin dan User.
3. Sistem tidak terhubung langsung dengan server operator seluler, melainkan hanya berupa simulasi program.
4. Sistem dibuat menggunakan bahasa pemrograman Python dan dijalankan melalui console atau terminal.
5. Sistem bersifat sederhana dan digunakan sebagai media pembelajaran.

1.4 Tujuan

Tujuan dari pembuatan dan penyusunan sistem ini adalah:

1. Membuat sistem yang dapat membantu pengelolaan data produk, pelanggan, dan transaksi secara digital.
2. Mempermudah pengguna dalam melihat dan membeli paket internet serta pulsa.
3. Memudahkan admin dalam menambah, memperbarui, dan menghapus data produk serta memantau laporan penjualan.
4. Mengimplementasikan konsep CRUD dan sistem multi-role (Admin dan User) menggunakan Python.
5. Meningkatkan kemampuan mahasiswa dalam merancang sistem informasi sederhana yang efisien.

1.5 Manfaat

Adapun manfaat dari sistem ini antara lain:

1. Bagi Admin, sistem membantu dalam mengelola data produk dan transaksi dengan lebih cepat serta akurat.
2. Bagi User, sistem memberikan kemudahan dalam melihat informasi produk dan melakukan pembelian dengan praktis.
3. Bagi Pelaku Usaha, sistem menjadi solusi digital untuk meningkatkan efisiensi dan efektivitas operasional.
4. Bagi Mahasiswa/Pengembang, sistem ini menjadi sarana pembelajaran dalam memahami konsep pemrograman dan sistem CRUD.
5. Bagi Konsumen, sistem memberikan pengalaman transaksi yang lebih cepat, mudah, dan tertata dengan baik.

BAB II
PERANCANGAN

2.1 Analisis Program

Program dengan judul **"Sistem Manajemen Paket Internet dan Pulsa FLASHCELL"** sangat bermanfaat dalam kehidupan sehari-hari, terutama di era digital saat ini, di mana hampir semua orang menggunakan layanan internet dan pulsa untuk berkomunikasi maupun mengakses berbagai kebutuhan online. Dengan adanya program ini, pengguna dapat melakukan transaksi pembelian paket internet dan pulsa dengan lebih mudah, cepat, dan efisien.

Aplikasi ini dirancang agar dapat membantu pengguna maupun admin dalam mengelola layanan secara sistematis dan terkomputerisasi. Program **"Sistem Manajemen Paket Internet dan Pulsa FLASHCELL"** juga mengikuti perkembangan zaman yang serba digital, sehingga seluruh proses transaksi dilakukan secara otomatis menggunakan sistem yang telah dibuat.

Analisis program ini dibuat untuk mempermudah proses transaksi, mencatat data pembelian, mengelola saldo pengguna, serta menyimpan seluruh aktivitas dalam file data yang rapi. Program ini juga mempermudah proses administrasi dengan penyimpanan data berbasis file sehingga semua aktivitas terekam dengan baik.

Berikut analisis mengenai kebutuhan dan komponen penyusun program:

1. Python

   Program ini dibuat menggunakan bahasa pemrograman *Python* karena mudah dipahami, sintaksnya sederhana, serta memiliki banyak pustaka pendukung yang memudahkan proses pengembangan program.

2. CSV (Comma Separated Values)

   Digunakan sebagai tempat penyimpanan data utama. File CSV berfungsi untuk menyimpan berbagai informasi seperti data pengguna, data produk (paket internet dan pulsa), serta riwayat transaksi. Format CSV dipilih karena sederhana dan mudah diakses oleh program.

3. PrettyTable Library

   Library ini digunakan untuk menampilkan data dalam bentuk tabel agar terlihat rapi dan mudah dibaca. Misalnya, untuk menampilkan daftar produk, daftar pengguna, maupun riwayat transaksi, semua ditampilkan dalam bentuk tabel yang terstruktur.

4. Pwinput Library

Digunakan untuk menyamarkan input password saat pengguna melakukan login. Dengan library ini, password yang diketik tidak akan terlihat di layar, sehingga keamanan akun lebih terjaga.

5. Colorama Library
   Digunakan untuk memberikan warna pada teks agar tampilan program lebih menarik dan interaktif. Misalnya, teks berwarna hijau untuk pesan sukses, merah untuk error, dan kuning untuk peringatan.

6. If, Elif, Else

   Struktur kontrol ini digunakan untuk membuat percabangan logika pada program.
   - **If** digunakan untuk mengecek kondisi tertentu, seperti validasi login.
   - **Elif** digunakan sebagai pengecekan kondisi lain jika kondisi pertama tidak terpenuhi.
   - **Else** dijalankan jika semua kondisi sebelumnya tidak sesuai.

7. Fungsi (def)

   Digunakan untuk membuat blok kode yang dapat digunakan berulang kali. Misalnya, `def login()` untuk proses login, `def register()` untuk pendaftaran akun baru, dan `def beli_paket()` untuk proses pembelian paket internet. Dengan fungsi, program menjadi lebih terstruktur dan mudah dikelola.

8. Loop “While True”

   Digunakan agar program dapat berjalan terus menerus sampai pengguna memutuskan untuk keluar. Misalnya, saat berada di menu utama, program akan terus menampilkan pilihan sampai pengguna memilih menu “Keluar”. Loop ini biasanya diakhiri dengan perintah `break` untuk menghentikan perulangan.

9. Try-Except

   Digunakan untuk menangani kesalahan input atau bug agar program tidak berhenti secara tiba-tiba. Contohnya, ketika pengguna menginput huruf pada kolom yang seharusnya angka, maka blok `try-except` akan menampilkan pesan kesalahan dan meminta input ulang tanpa menghentikan program.

10. Time dan Sys

    Digunakan untuk menambahkan efek animasi dan jeda waktu saat program berjalan, seperti efek loading atau saat menampilkan proses transaksi. Hal ini membuat tampilan program menjadi lebih interaktif dan realistis.

11. Datetime

    Library ini digunakan untuk menampilkan waktu dan tanggal saat transaksi dilakukan, serta untuk membuat ID transaksi yang unik berdasarkan waktu.

12. Otomatisasi Folder Penyimpanan Data

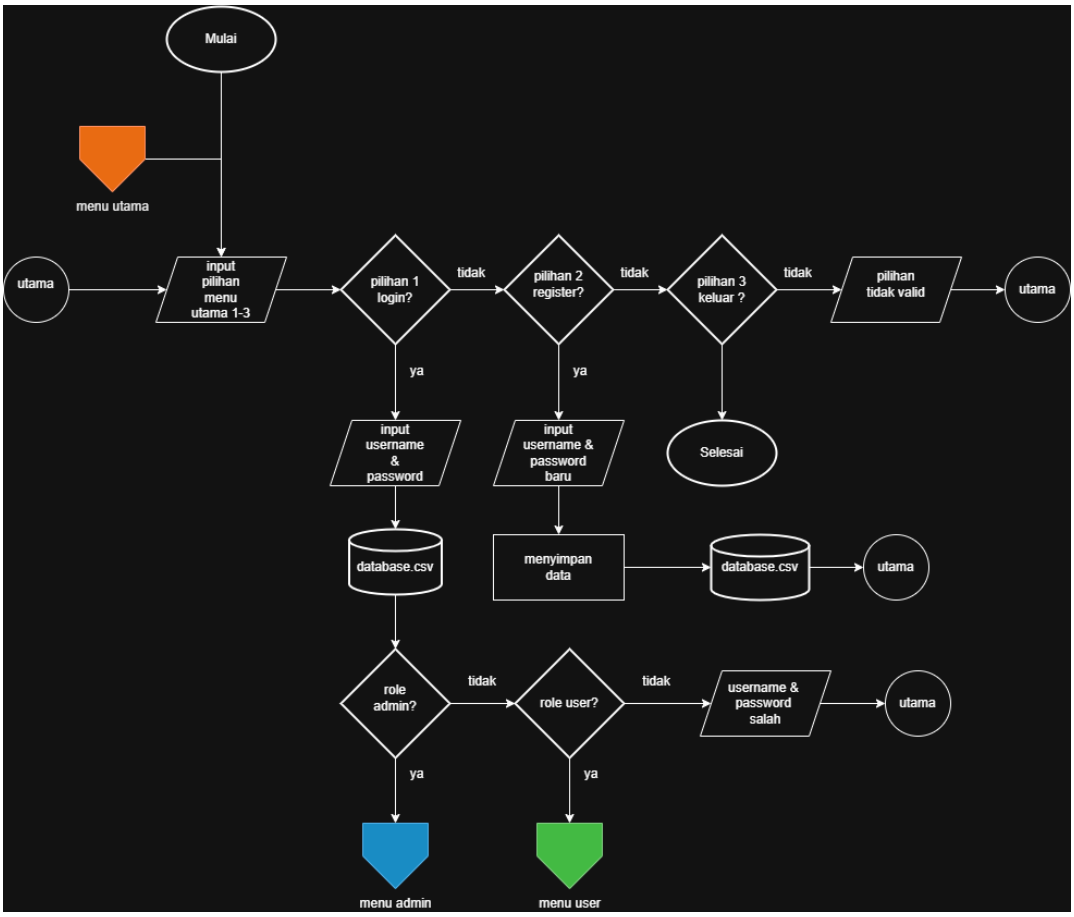
    Program ini juga dilengkapi dengan fitur **penyimpanan otomatis ke folder khusus**, sehingga setiap jenis data tersimpan di tempat yang berbeda agar lebih rapi dan mudah dikelola.
    Misalnya :

    - Folder **data/** untuk menyimpan file utama seperti data_pengguna.csv, data_produk.csv, dan riwayat_transaksi.csv.
    - Folder **struk/** digunakan untuk menyimpan hasil transaksi pembelian dalam bentuk file teks (.txt) yang berisi rincian pembelian, waktu, serta total pembayaran.
    - Jika folder belum ada, program akan otomatis membuat folder tersebut saat dijalankan pertama kali.

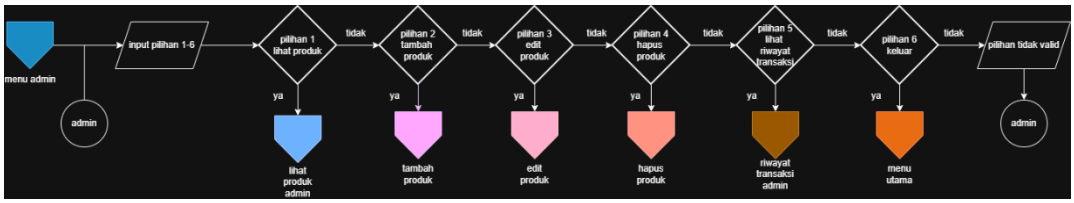
    Fitur ini sangat membantu dalam menjaga kerapian dan keamanan data agar tidak tercampur dengan file lain.

    Jadi, secara keseluruhan program ini dibuat dengan tujuan untuk mensimulasikan proses pembelian paket internet dan pulsa secara digital. Dengan adanya sistem ini, pengguna dapat bertransaksi secara mandiri dan efisien tanpa perlu proses manual, sedangkan admin dapat dengan mudah mengelola seluruh data produk dan transaksi melalui sistem yang telah terintegrasi.

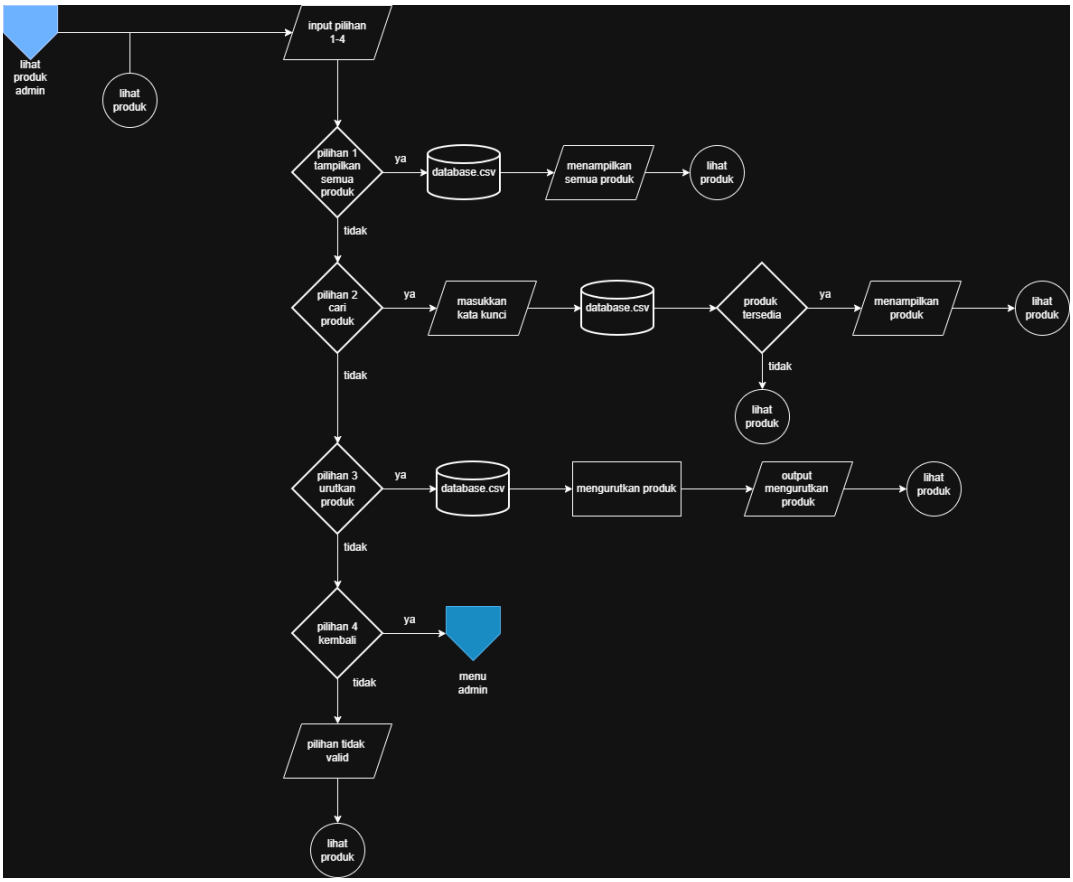
2.2 Flowchart



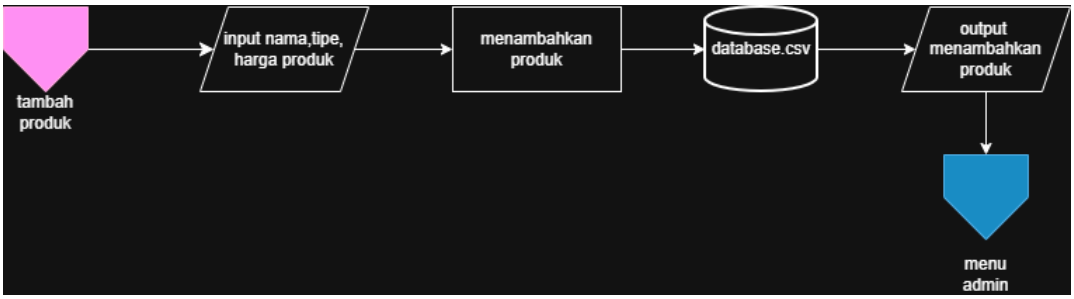
Gambar 2.1 Flowchart Menu Admin



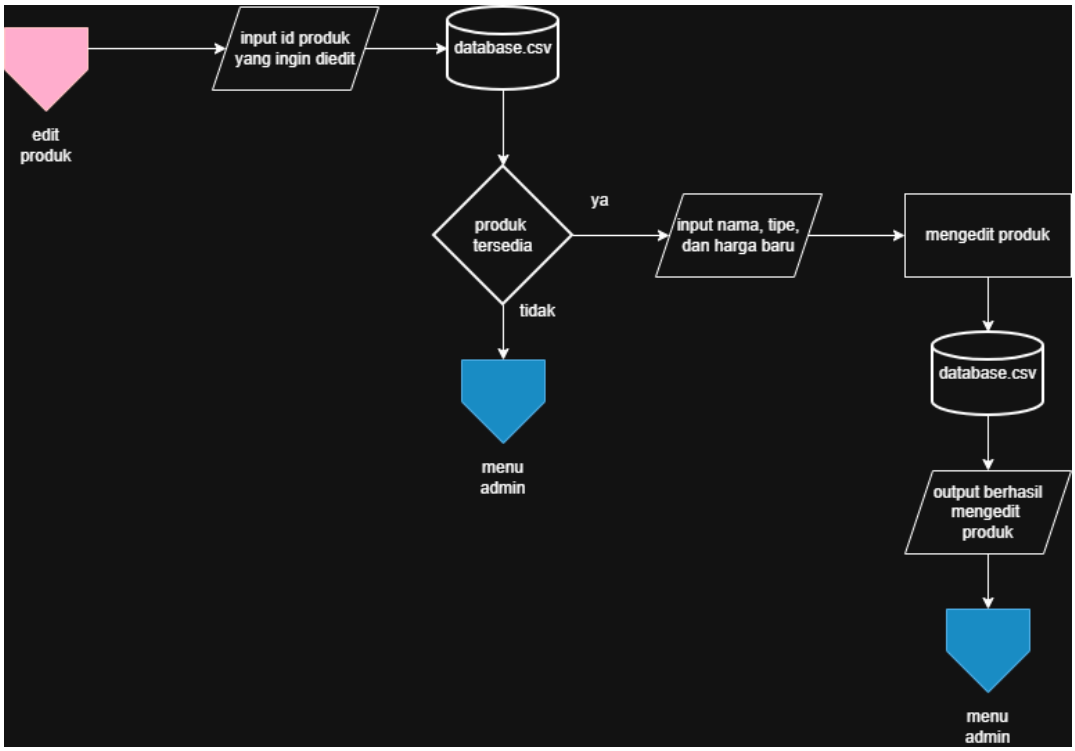
Gambar 2.2 Flowchart Menu Admin



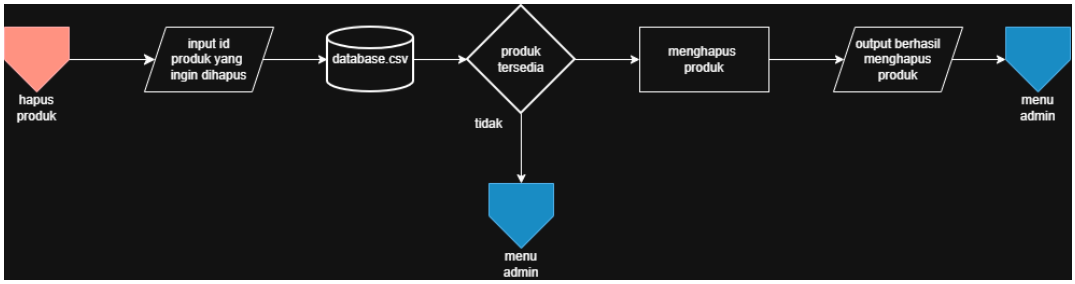
Gambar 2.3 Flowchart lihat produk admin



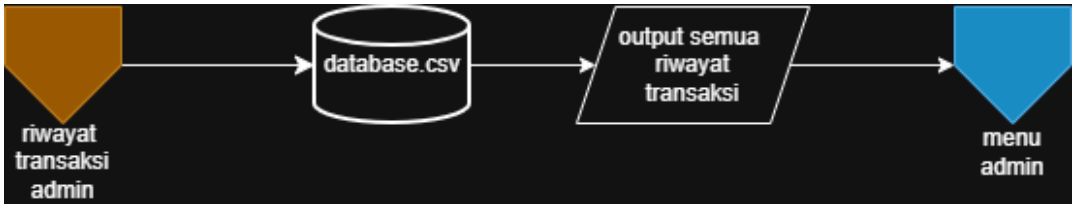
Gambar 2.4 Flowchart tambah produk admin



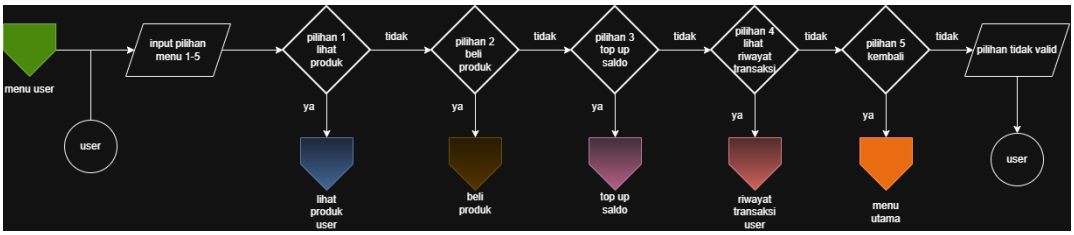
Gambar 2.5 Flowchart edit produk admin



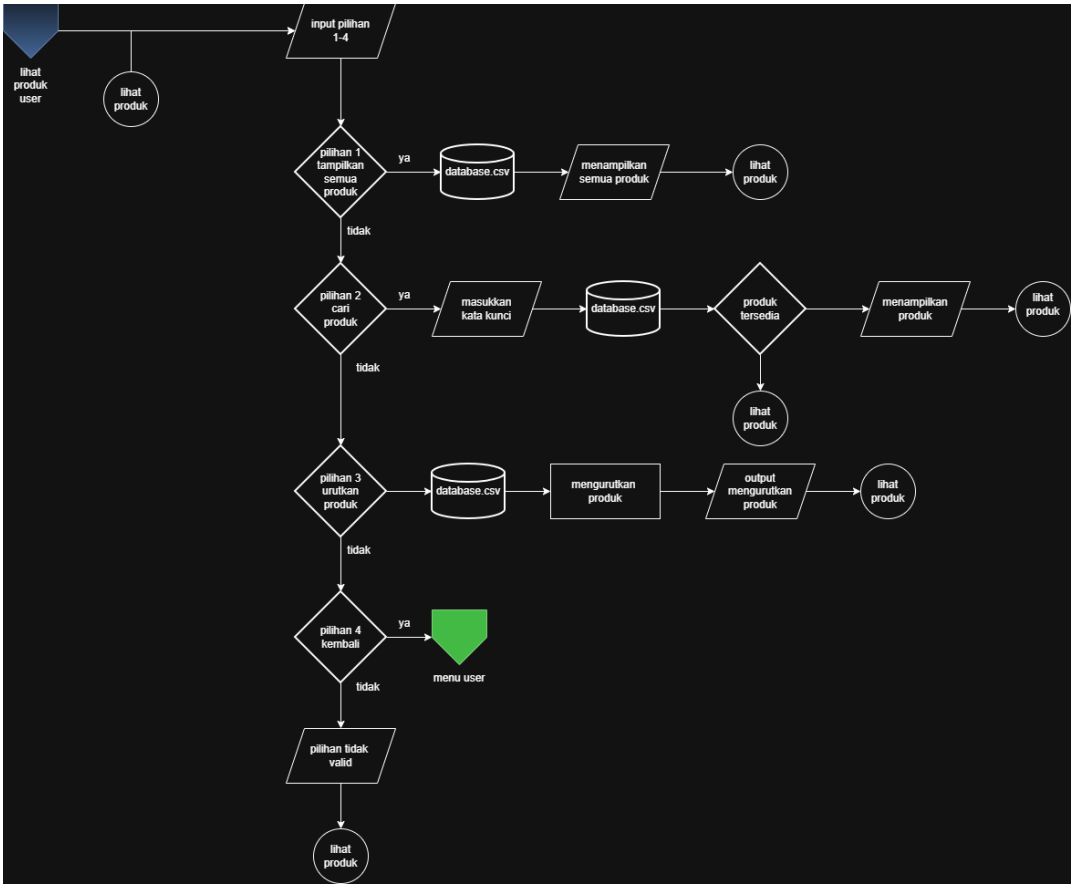
Gambar 2.6 Flowchart hapus produk admin



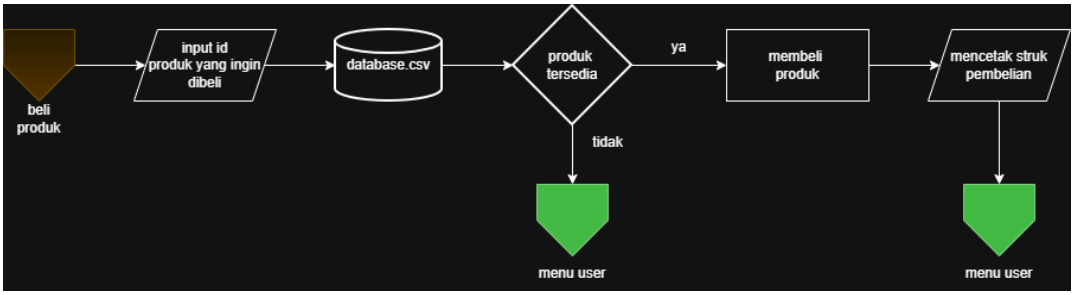
Gambar 2.7 Flowchart riwayat transaksi admin



Gambar 2.8 Flowchart menu user

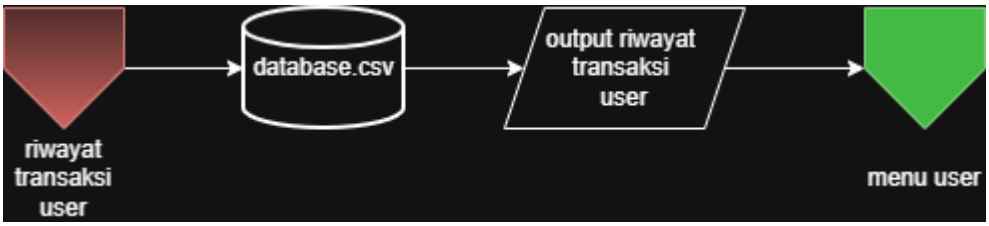


Gambar 2.9 Flowchart lihat produk user



Gambar 2.10 Flowchart beli produk user

Gambar 2.11 Flowchart top up saldo



Gambar 2.12 Flowchart lihat riwayat transaksi user

BAB III
HASIL DAN PEMBAHASAN

3.1 Implementasi Program

Program **"Sistem Manajemen Paket Internet dan Pulsa (FLASHCELL)"** dapat diimplementasikan secara nyata dalam kehidupan sehari-hari, khususnya di bidang **layanan digital dan penjualan pulsa online**.

Aplikasi ini dapat digunakan oleh pelaku usaha kecil seperti **konter pulsa**, **penjual paket data**, maupun **agen digital** untuk membantu proses transaksi secara lebih efisien dan terorganisir.

Berikut penjelasan implementasi masing-masing bagian program.

1. Python

   Program ini dibuat menggunakan **bahasa pemrograman Python**. Python dipilih karena memiliki sintaks yang sederhana, mudah dipelajari, serta banyak menyediakan pustaka (library) yang mendukung pengembangan sistem informasi.

2. Dictionary

   Struktur data **dictionary** digunakan dalam program ini untuk menyimpan data pengguna, data produk, dan data transaksi. Dictionary dipilih karena

```
headers_user = ("username", "password", "role", "saldo")
headers_produk = ("id", "nama", "tipe", "harga")
headers_transaksi = ("username", "nama_produk", "tipe", "harga", "waktu")
```

Gambar 3.1 Implementasi : Dictionary

   mampu menyimpan data dalam bentuk pasangan *key-value*, sehingga mempermudah pengambilan dan pengelolaan informasi.

Penjelasan:
Kode di atas mendefinisikan struktur data utama yang digunakan dalam sistem, yaitu data pengguna, data produk, dan data transaksi.

3. Import Module

   Program ini menggunakan beberapa *library* untuk mendukung jalannya sistem, antara lain:

   a. pwinput

   Berfungsi untuk menyamarkan input password ketika pengguna melakukan login atau registrasi agar tidak terlihat di layar.

   b. csv

Digunakan untuk membaca dan menulis data ke dalam file CSV yang berfungsi sebagai media penyimpanan utama data pengguna, produk, dan transaksi.

c. os
Berfungsi untuk membuat folder penyimpanan data secara otomatis, memeriksa keberadaan file, serta berinteraksi dengan sistem operasi.

d. time
Berfungsi untuk memberikan jeda waktu (*delay*) serta mencatat waktu transaksi pengguna.

e. sys
Digunakan untuk mengontrol eksekusi program, termasuk penghentian sistem secara manual melalui sys.exit().

f. datetime
Berfungsi untuk menampilkan waktu dan tanggal transaksi secara otomatis.

g. prettytable
Digunakan untuk menampilkan data dalam bentuk tabel agar tampilan terminal menjadi lebih rapi dan mudah dibaca.

h. colorama
Digunakan untuk memberikan efek warna pada teks di terminal, seperti pesan error, sukses, atau peringatan.

```
import os
import csv
import pwinput
import time
import sys
from datetime import datetime
from prettytable import PrettyTable
from colorama import Fore, Style, init
```

Gambar 3.2 Implementasi : Import Module

Penjelasan:
Kode di atas menunjukkan berbagai Library Python yang digunakan dalam program FLASHCELL untuk mengatur tampilan, penyimpanan data, dan interaksi pengguna.

4. CSV (Comma Separated Values)

   Program menggunakan file berformat CSV untuk menyimpan seluruh data secara dinamis. File CSV yang digunakan meliputi data_pengguna.csv, data_produk.csv, dan riwayat_transaksi.csv.

```
for file, header in [
    ("data/data_pengguna.csv", headers_user),
    ("data/data_produk.csv", headers_produk),
    ("data/riwayat_transaksi.csv", headers_transaksi)
]:
    if not os.path.exists(file):
        with open(file, "w", newline="", encoding="utf-8") as f:
            writer = csv.writer(f)
            writer.writerow(header)
```

Gambar 3.3 Implementasi : CSV

Penjelasan:
Kode tersebut memastikan bahwa folder dan file penyimpanan data dibuat secara otomatis ketika program pertama kali dijalankan.

5. Def (Function)

   Program ini menerapkan konsep **fungsi (def)** untuk meningkatkan modularitas dan efisiensi kode. Setiap fungsi memiliki tugas spesifik seperti membaca data, menyimpan data, login, registrasi, hingga transaksi

```
def load_csv(filename):
    try:
        with open(filename, "r", newline="", encoding="utf-8") as f:
            return list(csv.DictReader(f))
    except Exception:
        return []

def save_csv(filename, data, fieldnames):
    with open(filename, "w", newline="", encoding="utf-8") as f:
        writer = csv.DictWriter(f, fieldnames=fieldnames)
        writer.writeheader()
        writer.writerows(data)
```

Gambar 3.4 Implementasi : Def

Penjelasan:

Fungsi load _csv() digunakan untuk membaca data dari file CSV, sedangkan save_csv digunakan untuk menyimpan data. Kedua fungsi tersebut membantu agar pengelolaan data lebih efisien dan terstruktur.

6. If, Elif, Else

Struktur kendali **if, elif, else** digunakan untuk melakukan percabangan logika dalam program. Contohnya pada saat login, sistem akan memverifikasi username dan password.

```
def menu_user(username):
    while True:
        print(f"\n======== MENU USER ({username}) 👤 ========")
        print("1. Lihat Produk 🛍️")
        print("2. Beli Produk 🛒")
        print("3. Top Up Saldo 💸")
        print("4. Lihat Riwayat Transaksi 📜")
        print("5. Keluar 🔙")
        pilihan = input("Pilih menu: ").strip()
        if pilihan == "1":
            menu_lihat_produk()
        elif pilihan == "2":
            beli_produk(username)
        elif pilihan == "3":
            top_up_saldo(username)
        elif pilihan == "4":
            lihat_riwayat(username)
        elif pilihan == "5":
            break
        else:
            print(Fore.RED + "❌ Pilihan tidak valid!" + Style.RESET_ALL)
```

Gambar 3.5 Implementasi : If, Elif, Else

Penjelasan:

Dengan percabangan ini, program dapat menentukan tindakan berdasarkan kondisi tertentu, seperti menampilkan pesan keberhasilan atau kesalahan saat proses login.

7. While
   Struktur perulangan **while** digunakan untuk menampilkan menu utama secara terus-menerus selama pengguna belum memilih keluar dari sistem.

```
if __name__ == "__main__":
    buat_admin_default()
    buat_produk_default()

    try:
        while True:
            print("\n=== SISTEM MANAJEMEN PAKET INTERNET DAN PULSA FLASHCELL 💻 ===")
            print("1. Login 🔑")
            print("2. Register ✨")
            print("3. Keluar 🚪")
            pilihan = input("Pilih menu: ").strip()
            if pilihan == "1":
                user = login()
                if user:
                    if user.get("role") == "admin":
                        menu_admin()
                    else:
                        menu_user(user.get("username"))
            elif pilihan == "2":
                register()
            elif pilihan == "3":
                print("👋 Terima kasih telah berkunjung ke FLASHCELL! Sampai jumpa!")
                break
            else:
                print(Fore.RED + "❌ Pilihan tidak valid!" + Style.RESET_ALL)
```

Gambar 3.6 Implementasi : While

Penjelasan:
Perulangan ini memungkinkan pengguna melakukan berbagai aktivitas (login, register, transaksi) tanpa harus menjalankan ulang program.

8. CRUD (Create, Read, Update, Delete) If, Elif, Else
   Program FLASHCELL juga mengimplementasikan operasi CRUD dalam pengelolaan data produk.

a. Create

```python
def tambah_produk():
    data = load_csv("data/data_produk.csv")
    last_id = int(data[-1].get("id")) if data else 0
    id_produk = str(last_id + 1).zfill(3)
    nama = input("📝 Masukkan nama produk: ").strip()
    if not nama:
        print(Fore.RED + "❌ Nama produk tidak boleh kosong!" + Style.RESET_ALL)
        return

    while True:
        tipe = input("📦 Masukkan tipe (Internet/Pulsa): ").strip().title()
        if tipe in ["Internet", "Pulsa"]:
            break
        print(Fore.RED + "❌ Tipe produk harus 'Internet' atau 'Pulsa'." + Style.RESET_ALL)

    while True:
        harga_input = input("💰 Masukkan harga: ").strip()
        try:
            harga = int(harga_input)
            break
        except ValueError:
            print(Fore.RED + "❌ Harga harus berupa angka!" + Style.RESET_ALL)

    data.append({"id": id_produk, "nama": nama, "tipe": tipe, "harga": str(harga)})
    save_csv("data/data_produk.csv", data, headers_produk)
    print(Fore.GREEN + "✅ Produk berhasil ditambahkan!" + Style.RESET_ALL)
```

Gambar 3.7 Implementasi : Create

b. Read

```python
def tampilkan_produk(data=None):
    if data is None:
        data = load_csv("data/data_produk.csv")
    if not data:
        print(Fore.YELLOW + "⚠️ Belum ada data produk." + Style.RESET_ALL)
        return
    tabel = PrettyTable()
    tabel.field_names = ["ID", "Nama Produk", "Tipe", "Harga"]
    for p in data:
        tabel.add_row([p.get("id"), p.get("nama"), p.get("tipe"), f"Rp {p.get('harga')}"])
    print(tabel)
```

Gambar 3.8 Implementasi : Read

c. Update

```
def edit_produk():
    data = load_csv("data/data_produk.csv")
    tampilkan_produk(data)
    row = pilih_id_produk(data)
    if row:
        new_name = input(f"📝 Nama baru ({row.get('nama')}): ").strip()
        if new_name:
            row["nama"] = new_name

        tipe_input = input(f"📦 Tipe baru ({row.get('tipe')}): ").strip()
        if tipe_input:
            tipe_input = tipe_input.title()
            if tipe_input in ["Internet", "Pulsa"]:
                row["tipe"] = tipe_input
            else:
                print(Fore.RED + "❌ Tipe tidak valid. Tetap menggunakan tipe lama." + Style.RESET_ALL)

        harga_input = input(f"💰 Harga baru ({row.get('harga')}): ").strip()
        if harga_input:
            try:
                row["harga"] = str(int(harga_input))
            except ValueError:
                print(Fore.RED + "❌ Harga harus berupa angka! Tetap menggunakan harga lama." + Style.RESET_ALL)

        save_csv("data/data_produk.csv", data, headers_produk)
        print(Fore.GREEN + "✅ Produk berhasil diedit!" + Style.RESET_ALL)
```

Gambar 3.9 Implementasi : Update

d. Delete

```
def hapus_produk():
    data = load_csv("data/data_produk.csv")
    tampilkan_produk(data)
    row = pilih_id_produk(data)
    if row:
        confirm = input(f"❗ Yakin ingin menghapus produk '{row.get('nama')}'? (y/n): ").strip().lower()
        if confirm == "y":
            data.remove(row)
            # reset id
            for idx in range(len(data)):
                data[idx]["id"] = str(idx + 1).zfill(3)
            save_csv("data/data_produk.csv", data, headers_produk)
            print(Fore.GREEN + "✅ Produk dihapus dan ID diperbarui!" + Style.RESET_ALL)
        else:
            print(Fore.YELLOW + "⚠️ Hapus dibatalkan." + Style.RESET_ALL)
```

Gambar 3.10 Implementasi : Delete

Penjelasan:
Empat fungsi di atas menunjukkan implementasi fitur CRUD (Create, Read, Update, Delete) yang digunakan untuk mengelola data produk pada sistem.

9. Try–Except (Error Handling)
   Error Handling adalah mekanisme untuk **menangani kesalahan yang terjadi saat program berjalan** agar program tidak langsung crash, tetap berjalan aman, dan memberi pesan jelas kepada user.

```
    while True:
        harga_input = input("💰 Masukkan harga: ").strip()
        try:
            harga = int(harga_input)
            break
        except ValueError:
            print(Fore.RED + "❌ Harga harus berupa angka!" + Style.RESET_ALL)
```

Gambar 3.12 Implementasi : ValueError

```
        if pilihan == "1":
            user = login()
            if user:
                if user.get("role") == "admin":
                    menu_admin()
                else:
                    menu_user(user.get("username"))
        elif pilihan == "2":
            register()
        elif pilihan == "3":
            print("👋 Terima kasih telah berkunjung ke FLASHCELL! Sampai jumpa!")
            break
        else:
            print(Fore.RED + "❌ Pilihan tidak valid!" + Style.RESET_ALL)
except KeyboardInterrupt:
    print("\n❗ Program dihentikan. Sampai jumpa! 👋")
```

Gambar 3.11 Implementasi : KeyboardInterrupt

Penjelasan:

ValueError: Menangani kesalahan saat mengubah input user menjadi angka. KeyboardInterrupt: Menangani saat user menghentikan program paksa dengan menekan Ctrl+C.

3.2 Alur Program

Bagian ini menjelaskan urutan jalannya program *Sistem Manajemen Paket Internet dan Pulsa (FLASHCELL)* dari awal pengguna membuka program hingga proses logout. Setiap tahapan disertai potongan kode program, hasil tampilan di terminal, serta penjelasan terkait fungsi logika dari kode tersebut

1. Menu Utama

```
try:
    while True:
        print("\n=== SISTEM MANAJEMEN PAKET INTERNET DAN PULSA FLASHCELL 💻 ===")
        print("1. Login 🔑")
        print("2. Register ✨")
        print("3. Keluar 🚪")
        pilihan = input("Pilih menu: ").strip()
        if pilihan == "1":
            user = login()
            if user:
                if user.get("role") == "admin":
                    menu_admin()
                else:
                    menu_user(user.get("username"))
        elif pilihan == "2":
            register()
        elif pilihan == "3":
            print("👋 Terima kasih telah berkunjung ke FLASHCELL! Sampai jumpa!")
            break
        else:
            print(Fore.RED + "❌ Pilihan tidak valid!" + Style.RESET_ALL)
```

Gambar 3.13 Alur Program: Menu Utama

```
=== SISTEM MANAJEMEN PAKET INTERNET DAN PULSA FLASHCELL 💻 ===
1. Login 🔑
2. Register ✨
3. Keluar 🚪
Pilih menu:
```

Gambar 3.14 Hasil Terminal

Penjelasan:
Potongan kode di atas merupakan menu utama program. Pengguna diberikan tiga pilihan utama, yaitu **login, register, dan keluar**. Program menggunakan struktur **while True** agar menu terus ditampilkan selama pengguna belum memilih keluar dari sistem.

2. Registrasi Pengguna Baru

```
def register():
    users = load_csv("data/data_pengguna.csv")
    username = input(Fore.CYAN + "✨ Masukkan username baru: " + Style.RESET_ALL).strip()
    if not username:
        print(Fore.RED + "❌ Username tidak boleh kosong!" + Style.RESET_ALL)
        return
    for user in users:
        if user["username"] == username:
            print(Fore.RED + "❌ Username sudah terdaftar!" + Style.RESET_ALL)
            return
    password = pwinput.pwinput(Fore.CYAN + "🔒 Masukkan password baru: " + Style.RESET_ALL)
    users.append({"username": username, "password": password, "role": "user", "saldo": "200000"})
    save_csv("data/data_pengguna.csv", users, headers_user)
    print(Fore.GREEN + "✅ Registrasi berhasil! Saldo awal: Rp 200000 🎉" + Style.RESET_ALL)
```

Gambar 3.15 Alur Program: Registrasi Pengguna Baru

Penjelasan:
Pada proses registrasi, pengguna diminta memasukkan username dan password baru. Jika username belum terdaftar, sistem akan menyimpannya ke dalam file data_pengguna.csv dengan saldo awal Rp 200.000.

3. Login ke Sistem

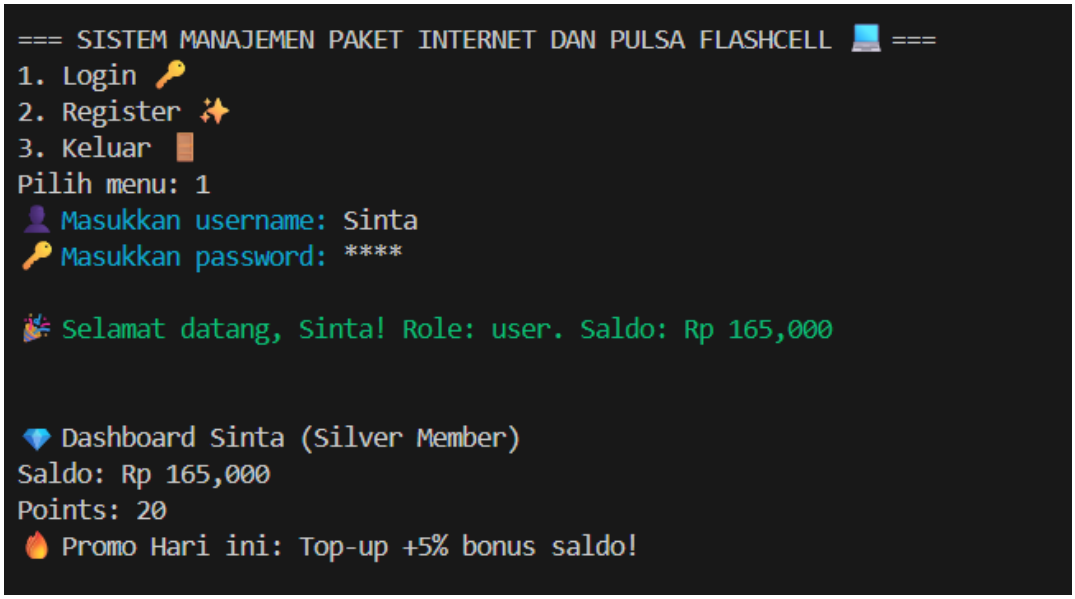
```
def login():
    users = load_csv("data/data_pengguna.csv")
    username = input(Fore.CYAN + "👤 Masukkan username: " + Style.RESET_ALL).strip()
    password = pwinput.pwinput(Fore.CYAN + "🔑 Masukkan password: " + Style.RESET_ALL)
    for user in users:
        if user["username"] == username and user["password"] == password:
            print(Fore.GREEN + f"\n🎉 Selamat datang, {username}! Role: {user['role']}. Saldo: Rp {user['saldo']}\n" + Style.RESET_ALL)
            return user
    print(Fore.RED + "❌ Login gagal! Username atau password salah." + Style.RESET_ALL)
    return None
```

Gambar 3.17 Alur Program: Login ke Sistem

```
=== SISTEM MANAJEMEN PAKET INTERNET DAN PULSA FLASHCELL 💻 ===
1. Login 🔑
2. Register ✨
3. Keluar 🚪
Pilih menu: 1
👤 Masukkan username: admin
🔑 Masukkan password: ***

🎉 Selamat datang, admin! Role: admin. Saldo: Rp 1000000
```

Gambar 3.18 Hasil Terminal Admin



Gambar 3.19 Hasil Terminal User

Penjelasan:
Proses login digunakan untuk memverifikasi identitas pengguna. Sistem membaca data dari data_pengguna.csv dan mencocokkan username serta password yang dimasukkan. Jika cocok, pengguna diarahkan ke menu sesuai perannya (admin atau user).

4. Menu Admin

```python
def menu_admin():
    while True:
        print("\n======== MENU ADMIN FLASHCELL 👑 ========")
        print("1. Lihat Produk 🛍️")
        print("2. Tambah Produk ➕")
        print("3. Edit Produk 🖍️")
        print("4. Hapus Produk 🗑️")
        print("5. Lihat Semua Riwayat Transaksi 📜")
        print("6. Keluar 🔙")
        pilihan = input("Pilih menu: ").strip()
        if pilihan == "1":
            menu_lihat_produk()
        elif pilihan == "2":
            tambah_produk()
        elif pilihan == "3":
            edit_produk()
        elif pilihan == "4":
            hapus_produk()
        elif pilihan == "5":
            lihat_semua_riwayat()
        elif pilihan == "6":
            break
        else:
            print(Fore.RED + "❌ Pilihan tidak valid!" + Style.RESET_ALL)
```

Gambar 3.20 Alur Program: Menu Admin

```
======== MENU ADMIN FLASHCELL 👑 ========
1. Lihat Produk 🛍️
2. Tambah Produk ➕
3. Edit Produk 🖍️
4. Hapus Produk 🗑️
5. Lihat Semua Riwayat Transaksi 📜
6. Keluar 🔙
Pilih menu:
```

Gambar 3.21 Hasil Terminal

Penjelasan:
Menu ini hanya dapat diakses oleh pengguna dengan peran admin. Admin memiliki akses penuh untuk mengelola data produk serta melihat semua transaksi pengguna.

a. Lihat Produk

```python
def menu_lihat_produk():
    data = load_csv("data/data_produk.csv")
    while True:
        print("\n---- MENU LIHAT PRODUK ----")
        print("1. Tampilkan semua produk 🛍️")
        print("2. Cari produk 🔍")
        print("3. Urutkan produk 📊")
        print("4. Kembali 🔙")
        pilihan = input("Pilih: ").strip()

        if pilihan == "1":
            tampilkan_produk(data)
        elif pilihan == "2":
            hasil = cari_produk_interaktif(data)
            if hasil:
                tampilkan_produk(hasil)
            else:
                print(Fore.YELLOW + "⚠️ Tidak ada produk yang cocok." + Style.RESET_ALL)
        elif pilihan == "3":
            data_sorted = urutkan_produk_interaktif(data)
            if data_sorted is not None:
                tampilkan_produk(data_sorted)
        elif pilihan == "4":
            break
        else:
            print(Fore.RED + "❌ Pilihan tidak valid!" + Style.RESET_ALL)
```

Gambar 3.22 Alur Program: Admin – Lihat Produk

```
---- MENU LIHAT PRODUK ----
1. Tampilkan semua produk 🛍️
2. Cari produk 🔍
3. Urutkan produk 📊
4. Kembali 🔙
Pilih: 1
+-----+-------------------------+----------+----------+
|  ID |       Nama Produk       |   Tipe   |  Harga   |
+-----+-------------------------+----------+----------+
| 001 |    Paket Internet 5GB   | Internet | Rp 20000 |
| 002 |   Paket Internet 10GB   | Internet | Rp 30000 |
| 003 |       Pulsa 25000       |  Pulsa   | Rp 25000 |
| 004 |       Pulsa 50000       |  Pulsa   | Rp 50000 |
| 005 | Paket Unlimited 1 Hari  | Internet | Rp 10000 |
+-----+-------------------------+----------+----------+
```

Gambar 3.23 Hasil Terminal

Penjelasan:

Pada menu **Lihat Produk**, admin dapat menampilkan daftar produk, melakukan pencarian, atau mengurutkan data produk berdasarkan harga atau nama. Program ini menggunakan **library PrettyTable** untuk menampilkan data produk dalam bentuk tabel yang lebih terstruktur dan mudah dibaca di terminal.

b. Tambah Produk

```
def tambah_produk():
    data = load_csv("data/data_produk.csv")
    last_id = int(data[-1].get("id")) if data else 0
    id_produk = str(last_id + 1).zfill(3)
    nama = input("📝 Masukkan nama produk: ").strip()
    if not nama:
        print(Fore.RED + "❌ Nama produk tidak boleh kosong!" + Style.RESET_ALL)
        return

    while True:
        tipe = input("📦 Masukkan tipe (Internet/Pulsa): ").strip().title()
        if tipe in ["Internet", "Pulsa"]:
            break
        print(Fore.RED + "❌ Tipe produk harus 'Internet' atau 'Pulsa'." + Style.RESET_ALL)

    while True:
        harga_input = input("💰 Masukkan harga: ").strip()
        try:
            harga = int(harga_input)
            break
        except ValueError:
            print(Fore.RED + "❌ Harga harus berupa angka!" + Style.RESET_ALL)

    data.append({"id": id_produk, "nama": nama, "tipe": tipe, "harga": str(harga)})
    save_csv("data/data_produk.csv", data, headers_produk)
    print(Fore.GREEN + "✅ Produk berhasil ditambahkan!" + Style.RESET_ALL)
```

Gambar 3.24 Alur Program: Admin – Tambah Produk

```
======== MENU ADMIN FLASHCELL 👑 ========
1. Lihat Produk 🛍️
2. Tambah Produk ➕
3. Edit Produk 🖍️
4. Hapus Produk 🗑️
5. Lihat Semua Riwayat Transaksi 📜
6. Keluar 🔙
Pilih menu: 2
📝 Masukkan nama produk: Paket Internet Unlimited 2 Hari
📦 Masukkan tipe (Internet/Pulsa): Internet
💰 Masukkan harga: 13000
✅ Produk berhasil ditambahkan!
```

Gambar 3.25 Hasil Terminal

Penjelasan:
Fungsi ini digunakan untuk menambah produk baru ke dalam sistem. Admin diminta

mengisi nama produk, tipe (Internet atau Pulsa), serta harga produk. Data produk kemudian disimpan ke file data_produk.csv. Terdapat pula **error handling (ValueError)** untuk memastikan bahwa input harga harus berupa angka.

c. Edit Produk

```
def edit_produk():
    data = load_csv("data/data_produk.csv")
    tampilkan_produk(data)
    row = pilih_id_produk(data)
    if row:
        new_name = input(f"📝 Nama baru ({row.get('nama')}): ").strip()
        if new_name:
            row["nama"] = new_name

        tipe_input = input(f"📦 Tipe baru ({row.get('tipe')}): ").strip()
        if tipe_input:
            tipe_input = tipe_input.title()
            if tipe_input in ["Internet", "Pulsa"]:
                row["tipe"] = tipe_input
            else:
                print(Fore.RED + "❌ Tipe tidak valid. Tetap menggunakan tipe lama." + Style.RESET_ALL)

        harga_input = input(f"💰 Harga baru ({row.get('harga')}): ").strip()
        if harga_input:
            try:
                row["harga"] = str(int(harga_input))
            except ValueError:
                print(Fore.RED + "❌ Harga harus berupa angka! Tetap menggunakan harga lama." + Style.RESET_ALL)

        save_csv("data/data_produk.csv", data, headers_produk)
        print(Fore.GREEN + "✅ Produk berhasil diedit!" + Style.RESET_ALL)
```

Gambar 3.26 Alur Program: Admin – Edit Produk

```
======== MENU ADMIN FLASHCELL 👑 ========
1. Lihat Produk 🛍️
2. Tambah Produk ➕
3. Edit Produk 🖍️
4. Hapus Produk 🗑️
5. Lihat Semua Riwayat Transaksi 📜
6. Keluar 🔙
Pilih menu: 3
+-----+----------------------------------+----------+----------+
|  ID |           Nama Produk            |   Tipe   |  Harga   |
+-----+----------------------------------+----------+----------+
| 001 |        Paket Internet 5GB        | Internet | Rp 20000 |
| 002 |       Paket Internet 10GB        | Internet | Rp 30000 |
| 003 |           Pulsa 25000            |  Pulsa   | Rp 25000 |
| 004 |           Pulsa 50000            |  Pulsa   | Rp 50000 |
| 005 |      Paket Unlimited 1 Hari      | Internet | Rp 10000 |
| 006 | Paket Internet Unlimited 2 Hari  | Internet | Rp 13000 |
+-----+----------------------------------+----------+----------+
🆔 Masukkan ID produk: 005
📝 Nama baru (Paket Unlimited 1 Hari): Pulsa 55000
📦 Tipe baru (Internet): Pulsa
💰 Harga baru (10000): 55000
✅ Produk berhasil diedit!
```

Penjelasan:

Fitur **Edit Produk** memungkinkan admin memperbarui data produk yang sudah ada, seperti nama, tipe, dan harga.

d. Delete Produk

```python
def hapus_produk():
    data = load_csv("data/data_produk.csv")
    tampilkan_produk(data)
    row = pilih_id_produk(data)
    if row:
        confirm = input(f"❗ Yakin ingin menghapus produk '{row.get('nama')}'? (y/n): ").strip().lower()
        if confirm == "y":
            data.remove(row)
            # reset id
            for idx in range(len(data)):
                data[idx]["id"] = str(idx + 1).zfill(3)
            save_csv("data/data_produk.csv", data, headers_produk)
            print(Fore.GREEN + "✅ Produk dihapus dan ID diperbarui!" + Style.RESET_ALL)
        else:
            print(Fore.YELLOW + "⚠️ Hapus dibatalkan." + Style.RESET_ALL)
```

Gambar 3.28 Alur Program: Admin – Hapus Produk

```
======== MENU ADMIN FLASHCELL 👑 ========
1. Lihat Produk 🛍️
2. Tambah Produk ➕
3. Edit Produk 🖍️
4. Hapus Produk 🗑️
5. Lihat Semua Riwayat Transaksi 📜
6. Keluar 🔙
Pilih menu: 4
+-----+----------------------------------+----------+----------+
|  ID |           Nama Produk            |   Tipe   |  Harga   |
+-----+----------------------------------+----------+----------+
| 001 |        Paket Internet 5GB        | Internet | Rp 20000 |
| 002 |       Paket Internet 10GB        | Internet | Rp 30000 |
| 003 |           Pulsa 25000            |  Pulsa   | Rp 25000 |
| 004 |           Pulsa 50000            |  Pulsa   | Rp 50000 |
| 005 |      Paket Unlimited 1 Hari      | Internet | Rp 10000 |
| 006 | Paket Internet Unlimited 2 Hari  | Internet | Rp 13000 |
+-----+----------------------------------+----------+----------+
🆔 Masukkan ID produk: 006
❗ Yakin ingin menghapus produk 'Paket Internet Unlimited 2 Hari'? (y/n): y
✅ Produk dihapus dan ID diperbarui!
```

Gambar 3.29 Hasil Terminal

Penjelasan:

Fitur ini digunakan untuk menghapus produk yang sudah tidak tersedia. Setelah produk dihapus, sistem akan memperbarui ID agar tetap berurutan. Proses ini juga menggunakan **konfirmasi (y/n)** untuk mencegah penghapusan yang tidak disengaja.

e. Lihat Semua Riwayat Transaksi

```python
def lihat_semua_riwayat():
    data_transaksi = load_csv("data/riwayat_transaksi.csv")
    if not data_transaksi:
        print(Fore.YELLOW + "\n⚠️ Belum ada riwayat transaksi.\n" + Style.RESET_ALL)
        return
    tabel_transaksi = PrettyTable()
    tabel_transaksi.field_names = ["Username", "Nama Produk", "Tipe", "Harga", "Waktu"]
    for trx in data_transaksi:
        tabel_transaksi.add_row([trx.get("username"), trx.get("nama_produk"), trx.get("tipe"), f"Rp {trx.get('harga')}", trx.get("waktu")])
    print(tabel_transaksi)
```

Gambar 3.30 Alur Program: Admin – Riwayat Transaksi

```
======== MENU ADMIN FLASHCELL 👑 ========
1. Lihat Produk 🛍️
2. Tambah Produk ➕
3. Edit Produk ✏️
4. Hapus Produk 🗑️
5. Lihat Semua Riwayat Transaksi 📜
6. Keluar 🔙
Pilih menu: 5
+----------+------------------------+----------+----------+---------------------+
| Username |      Nama Produk       |   Tipe   |  Harga   |        Waktu        |
+----------+------------------------+----------+----------+---------------------+
|  Sinta   | Paket Unlimited 1 Hari | Internet | Rp 10000 | 2025-10-20 20:27:54 |
|  Sinta   |      Pulsa 25000       |  Pulsa   | Rp 25000 | 2025-10-20 20:28:17 |
+----------+------------------------+----------+----------+---------------------+
```

Gambar 3.31 Hasil Terminal

Penjelasan:
Menu ini menampilkan seluruh riwayat transaksi yang dilakukan oleh pengguna. Admin dapat memantau aktivitas pembelian produk, lengkap dengan nama pengguna, jenis produk, harga, dan waktu transaksi.

f. Logout

```
======== MENU ADMIN FLASHCELL 👑 ========
1. Lihat Produk 🛍️
2. Tambah Produk ➕
3. Edit Produk ✏️
4. Hapus Produk 🗑️
5. Lihat Semua Riwayat Transaksi 📜
6. Logout 🔐
Pilih menu: 6

🔐 Anda telah logout dari akun Admin 👑
🏠 Kembali ke menu utama FLASHCELL 💻
```

Gambar 3.32 Logout Dari Menu Admin

Penjelasan:
Setelah menyelesaikan pengelolaan data, admin dapat keluar dari sistem dengan memilih opsi logout. Program akan mengembalikan admin ke menu utama untuk memilih tindakan selanjutnya.

g. Searching Produk

```
def cari_produk_interaktif(data):
    keyword = input("🔎 Masukkan kata kunci (nama/tipe/harga): ").strip().lower()
    hasil = [p for p in data if keyword in p.get("nama", "").lower() or keyword in p.get("tipe", "").lower() or keyword in p.get("harga", "")]
    return hasil
```

Gambar 3.33 Alur Program: Admin – Searching Produk

```
---- MENU LIHAT PRODUK ----
1. Tampilkan semua produk
2. Cari produk
3. Urutkan produk
4. Kembali
Pilih: 2
🔎 Masukkan kata kunci (nama/tipe/harga): Pulsa
+-----+-------------+-------+----------+
|  ID | Nama Produk |  Tipe |  Harga   |
+-----+-------------+-------+----------+
| 003 | Pulsa 25000 | Pulsa | Rp 25000 |
| 004 | Pulsa 50000 | Pulsa | Rp 50000 |
+-----+-------------+-------+----------+
```

Gambar 3.34 Hasil Terminal

Penjelasan:
Fitur *searching* atau pencarian ini memungkinkan admin mencari data produk berdasarkan nama, tipe, atau harga. Proses pencarian dilakukan secara interaktif dengan menerima input kata kunci dari pengguna. Setiap produk yang mengandung kata kunci tersebut akan ditampilkan menggunakan **PrettyTable**, sehingga hasil pencarian terlihat rapi.

h. Sorting Produk

```python
def urutkan_produk_interaktif(data):
    print("\nPilihan pengurutan:")
    print("1. Harga (Termurah → Termahal)")
    print("2. Harga (Termahal → Termurah)")
    print("3. Nama (A-Z)")
    print("4. Nama (Z-A)")
    opsi = input("Pilih: ").strip()
    if opsi == "1":
        data_sorted = sorted(data, key=lambda x: int(x.get("harga", 0)))
    elif opsi == "2":
        data_sorted = sorted(data, key=lambda x: int(x.get("harga", 0)), reverse=True)
    elif opsi == "3":
        data_sorted = sorted(data, key=lambda x: x.get("nama", "").lower())
    elif opsi == "4":
        data_sorted = sorted(data, key=lambda x: x.get("nama", "").lower(), reverse=True)
    else:
        print(Fore.RED + "❌ Pilihan tidak valid!" + Style.RESET_ALL)
        return None
    return data_sorted
```

Gambar 3.35 Program: Admin – Sorting Produk

```
---- MENU LIHAT PRODUK ----
1. Tampilkan semua produk
2. Cari produk
3. Urutkan produk
4. Kembali
Pilih: 3

Pilihan pengurutan:
1. Harga (Termurah → Termahal)
2. Harga (Termahal → Termurah)
3. Nama (A-Z)
4. Nama (Z-A)
Pilih: 1
+-----+------------------------+----------+----------+
|  ID |      Nama Produk       |   Tipe   |  Harga   |
+-----+------------------------+----------+----------+
| 005 | Paket Unlimited 1 Hari | Internet | Rp 10000 |
| 001 |   Paket Internet 5GB   | Internet | Rp 20000 |
| 003 |      Pulsa 25000       |  Pulsa   | Rp 25000 |
| 002 |  Paket Internet 10GB   | Internet | Rp 30000 |
| 004 |      Pulsa 50000       |  Pulsa   | Rp 50000 |
+-----+------------------------+----------+----------+
```

Gambar 3.36 Alur Program: Hasil Terminal

Penjelasan:
Fitur *sorting* digunakan untuk mengurutkan produk berdasarkan harga atau nama. Admin dapat memilih urutan data dari termurah hingga termahal, sebaliknya, atau

secara alfabetis (A–Z / Z–A). Fungsi ini memanfaatkan metode **sorted()** dengan **lambda expression** sebagai kunci pengurutan.

5. Menu User

```python
# ==================== MENU USER ====================
def menu_user(username):
    while True:
        print(f"\n======== MENU USER ({username}) 👤 ========")
        print("1. Lihat Produk 🛍️")
        print("2. Beli Produk 🛒")
        print("3. Top Up Saldo 💸")
        print("4. Lihat Riwayat Transaksi 📜")
        print("5. Logout 🔐")

        pilihan = input("Pilih menu: ").strip()

        if pilihan == "1":
            menu_lihat_produk()
        elif pilihan == "2":
            beli_produk(username)
        elif pilihan == "3":
            top_up_saldo(username)
        elif pilihan == "4":
            lihat_riwayat(username)
        elif pilihan == "5":
            print(Fore.CYAN + f"\n🔐 {username} telah logout 👋" + Style.RESET_ALL)
            print(Fore.YELLOW + "🏠 Kembali ke menu utama FLASHCELL 💻" + Style.RESET_ALL)
            time.sleep(1.5)
            break  # pakai break, bukan return
        else:
            print(Fore.RED + "❌ Pilihan tidak valid!" + Style.RESET_ALL)
```

Gambar 3.37 Alur Program: Menu User

```
======== MENU USER (Sinta) 👤 ========
1. Lihat Produk 🛍️
2. Beli Produk 🛒
3. Top Up Saldo 💸
4. Lihat Riwayat Transaksi 📜
5. Logout 🔐
Pilih menu:
```

Gambar 3.38 Hasil Terminal

Penjelasan:
Menu ini ditampilkan setelah pengguna berhasil login dengan role user. User dapat melakukan transaksi seperti melihat produk, membeli produk, melakukan top up saldo, serta melihat riwayat transaksi. Setiap pilihan menu memiliki fungsi yang berbeda dan dijalankan secara modular.

a. Lihat Produk

```python
def tampilkan_produk(data=None):
    if data is None:
        data = load_csv("data/data_produk.csv")
    if not data:
        print(Fore.YELLOW + "⚠️ Belum ada data produk." + Style.RESET_ALL)
        return
    tabel = PrettyTable()
    tabel.field_names = ["ID", "Nama Produk", "Tipe", "Harga"]
    for p in data:
        tabel.add_row([p.get("id"), p.get("nama"), p.get("tipe"), f"Rp {p.get('harga')}"])
    print(tabel)
```

Gambar 3.39 Alur Program: User – Lihat Produk

```
---- MENU LIHAT PRODUK ----
1. Tampilkan semua produk
2. Cari produk
3. Urutkan produk
4. Kembali
Pilih: 4

======== MENU USER (Sinta) ========
1. Lihat Produk
2. Beli Produk
3. Top Up Saldo
4. Lihat Riwayat Transaksi
5. Logout
Pilih menu: 1

---- MENU LIHAT PRODUK ----
1. Tampilkan semua produk
2. Cari produk
3. Urutkan produk
4. Kembali
Pilih: 1
+-----+-----------------------+----------+----------+
|  ID |      Nama Produk      |   Tipe   |  Harga   |
+-----+-----------------------+----------+----------+
| 001 |   Paket Internet 5GB  | Internet | Rp 20000 |
| 002 |  Paket Internet 10GB  | Internet | Rp 30000 |
| 003 |      Pulsa 25000      |  Pulsa   | Rp 25000 |
| 004 |      Pulsa 50000      |  Pulsa   | Rp 50000 |
| 005 | Paket Unlimited 1 Hari | Internet | Rp 10000 |
+-----+-----------------------+----------+----------+
```

Gambar 3.40 Hasil Terminal

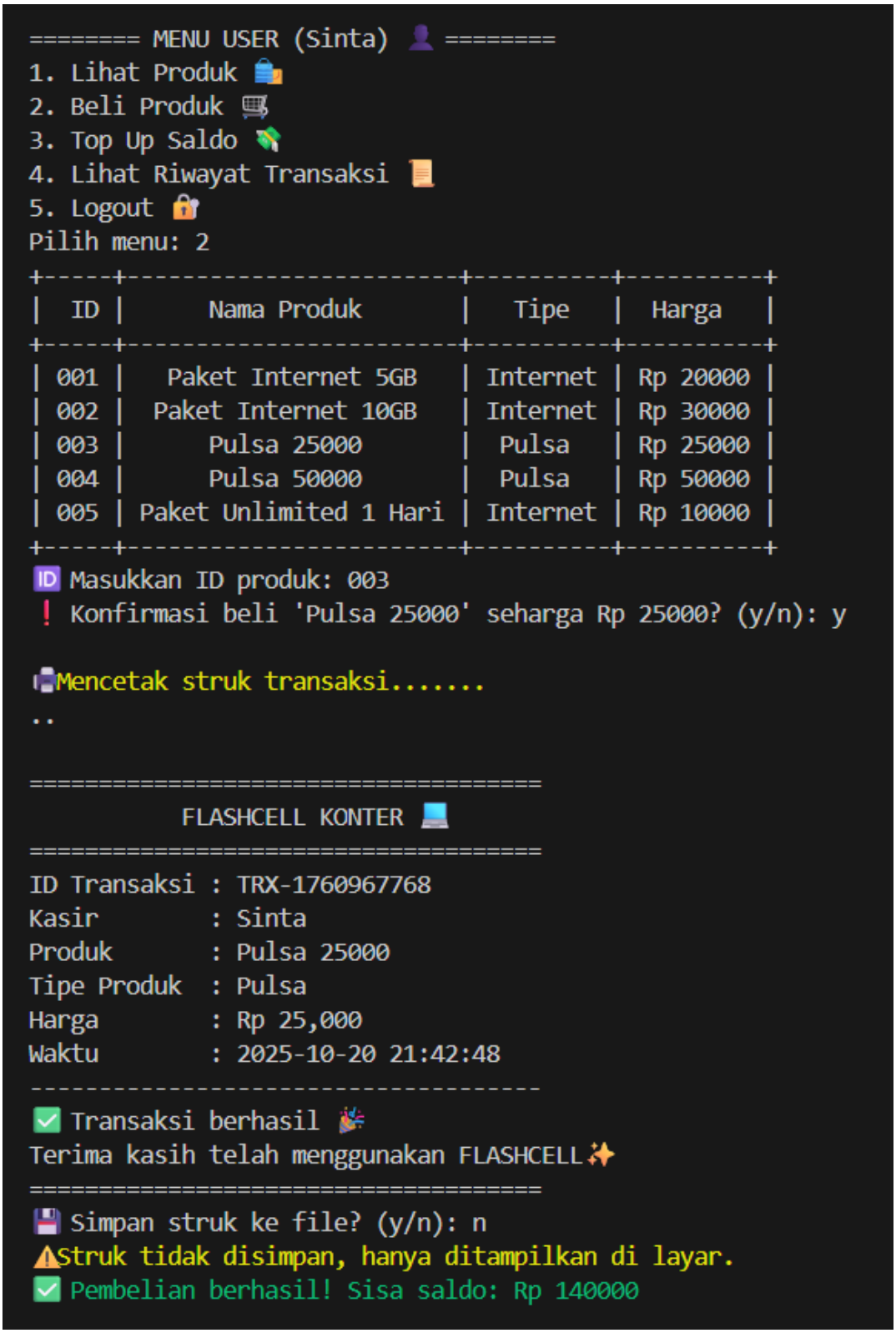
Penjelasan:
Pada menu ini, user dapat melihat seluruh daftar produk yang tersedia. Data produk

diambil dari file data_produk.csv an ditampilkan dengan tabel agar lebih terstruktur. Fitur ini sama seperti milik admin, namun tanpa akses untuk mengubah data produk.

b. Beli Produk

```
def beli_produk(username):
    data_produk = load_csv("data/data_produk.csv")
    tampilkan_produk(data_produk)
    produk_row = pilih_id_produk(data_produk)
    if not produk_row:
        return
    users = load_csv("data/data_pengguna.csv")
    for user in users:
        if user.get("username") == username:
            harga = int(produk_row.get("harga", 0))
            if int(user.get("saldo", 0)) >= harga:
                confirm = input(f" ! Konfirmasi beli '{produk_row.get('nama')}' seharga Rp {harga}? (y/n): ").strip().lower()
                if confirm != "y":
                    print(Fore.YELLOW + "⚠ Transaksi dibatalkan." + Style.RESET_ALL)
                    return
                user["saldo"] = str(int(user.get("saldo", 0)) - harga)
                save_csv("data/data_pengguna.csv", users, headers_user)
                waktu = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
                data_transaksi = load_csv("data/riwayat_transaksi.csv")
                data_transaksi.append({
                    "username": username,
                    "nama_produk": produk_row.get("nama"),
                    "tipe": produk_row.get("tipe"),
                    "harga": str(harga),
                    "waktu": waktu
                })
                save_csv("data/riwayat_transaksi.csv", data_transaksi, headers_transaksi)
                tampilkan_invoice(username, produk_row, harga, waktu)
                print(Fore.GREEN + f"✅ Pembelian berhasil! Sisa saldo: Rp {user.get('saldo')}" + Style.RESET_ALL)
                return
            else:
                print(Fore.RED + "❌ Saldo tidak cukup!" + Style.RESET_ALL)
                return
    print(Fore.RED + "❌ Akun tidak ditemukan!" + Style.RESET_ALL)
```

Gambar 3.41 Alur Program: User – Beli Produk



Gambar 3.42 Hasil Terminal

Penjelasan:
Fitur ini digunakan oleh user untuk membeli produk yang tersedia. Sistem akan mengecek apakah saldo mencukupi, lalu mengurangi saldo dan mencatat transaksi

ke file riwayat_transaksi.csv. Struk transaksi ditampilkan di layar dan dapat disimpan sebagai file .txt.

c. Top Up Saldo

```python
def top_up_saldo(username):
    users = load_csv("data/data_pengguna.csv")
    for user in users:
        if user.get("username") == username:
            percobaan = 0
            while percobaan < 3:
                try:
                    jumlah = int(input("💸 Masukkan jumlah top-up (min 1000, max 500000): ").strip())
                    if jumlah < 1000:
                        print(Fore.RED + "❌ Minimal top-up Rp 1.000" + Style.RESET_ALL)
                        percobaan += 1
                        continue
                    if jumlah > 500000:
                        print(Fore.RED + "❌ Maksimal top-up Rp 500.000 per transaksi" + Style.RESET_ALL)
                        percobaan += 1
                        continue
                    if int(user.get("saldo", 0)) + jumlah > 5000000:
                        print(Fore.RED + "❌ Saldo maksimal Rp 5.000.000. Top-up dibatalkan." + Style.RESET_ALL)
                        return
                    user["saldo"] = str(int(user.get("saldo", 0)) + jumlah)
                    save_csv("data/data_pengguna.csv", users, headers_user)
                    print(Fore.GREEN + f"✅ Top-up berhasil! Saldo baru: Rp {user['saldo']}" + Style.RESET_ALL)
                    return
                except ValueError:
                    print(Fore.RED + "❌ Harus berupa angka!" + Style.RESET_ALL)
                    percobaan += 1
            print(Fore.RED + "❌ Terlalu banyak kesalahan. Kembali ke menu sebelumnya." + Style.RESET_ALL)
            return
    print(Fore.RED + "❌ Akun tidak ditemukan!" + Style.RESET_ALL)
```

Gambar 3.43 Alur Program: User – Top Up Saldo

```
======== MENU USER (Sinta) 👤 ========
1. Lihat Produk 🛍️
2. Beli Produk 🛒
3. Top Up Saldo 💸
4. Lihat Riwayat Transaksi 📜
5. Logout 🔐
Pilih menu: 3
💸 Masukkan jumlah top-up (min 1000, max 500000): 100000
✅ Top-up berhasil! Saldo baru: Rp 240000
```

Gambar 3.44 Hasil Terminal

Penjelasan:
Fitur Top Up Saldo memungkinkan user menambah saldo akun. Program membatasi jumlah minimum dan maksimum top up, serta memeriksa agar saldo tidak melebihi Rp 5.000.000. Terdapat pula **error handling (try–except ValueError)** untuk mencegah input non-numerik.

d. Lihat Riwayat Transaksi

```python
def lihat_riwayat(username):
    data_transaksi = load_csv("data/riwayat_transaksi.csv")
    user_trx = [trx for trx in data_transaksi if trx.get("username")==username]
    if not user_trx:
        print(Fore.YELLOW + "\n⚠ Belum ada riwayat transaksi.\n" + Style.RESET_ALL)
        return
    tabel = PrettyTable()
    tabel.field_names = ["Nama Produk","Tipe","Harga","Masa Aktif","Waktu"]
    for trx in user_trx:
        tabel.add_row([
            trx.get("nama_produk"),
            trx.get("tipe"),
            format_rp(trx.get("harga")),
            trx.get("masa_aktif","0"),
            trx.get("waktu")
        ])
    print(tabel)
```

Gambar 3.45 Alur Program: User – Lihat Riwayat Transaksi

```
======== MENU USER (Sinta) 👤 ========
1. Lihat Produk 🛍️
2. Beli Produk 🛒
3. Top Up Saldo 💸
4. Lihat Riwayat Transaksi 📜
5. Keluar 🔙
Pilih menu: 4
+-----------------------+----------+-----------+------------+---------------------+
|      Nama Produk      |   Tipe   |   Harga   | Masa Aktif |       Waktu         |
+-----------------------+----------+-----------+------------+---------------------+
|      Pulsa 25000      |  Pulsa   | Rp 25,000 |     0      | 2025-10-26 12:02:25 |
| Paket Unlimited 1 Hari | Internet | Rp 10,000 |     1      | 2025-10-26 12:03:19 |
|   Paket Internet 5GB  | Internet | Rp 20,000 |     5      | 2025-10-26 16:02:10 |
+-----------------------+----------+-----------+------------+---------------------+
```

Gambar 3.46 Hasil Terminal

Penjelasan:
Menu ini digunakan untuk menampilkan riwayat transaksi yang dilakukan oleh user. Data riwayat diambil dari file riwayat_transaksi.csv dan disajikan dalam bentuk tabel. dan disajikan dalam bentuk tabel.

e. Logout

```
======== MENU USER (Sinta) 👤 ========
1. Lihat Produk 🛍️
2. Beli Produk 🛒
3. Top Up Saldo 💸
4. Lihat Riwayat Transaksi 📜
5. Logout 🔐
Pilih menu: 5

🔐 Sinta telah logout 👋
🏠 Kembali ke menu utama FLASHCELL 💻

=== SISTEM MANAJEMEN PAKET INTERNET DAN PULSA FLASHCELL 💻 ===
1. Login 🔑
2. Register ✨
3. Keluar 🚪
Pilih menu:
```

Gambar 3.47 Alur Program: User – Logout

Penjelasan:
Opsi Logout digunakan untuk keluar dari akun user dan kembali ke menu utama. Hal ini menandakan berakhirnya sesi pengguna tanpa harus menutup program secara keseluruhan.

6. Pemilihan ID Produk

```python
def pilih_id_produk(data, max_attempt=3):
    percobaan = 0
    while percobaan < max_attempt:
        id_produk = input("🆔 Masukkan ID produk: ").strip()
        for row in data:
            if row.get("id") == id_produk:
                return row
        percobaan += 1
        print(Fore.RED + f"❌ ID tidak ditemukan. Sisa percobaan: {max_attempt-percobaan}" + Style.RESET_ALL)
    print(Fore.RED + "❌ Terlalu banyak kesalahan. Kembali ke menu sebelumnya." + Style.RESET_ALL)
    return None
```

Gambar 3.48 Alur Program: Pemilihan ID Produk

```
🆔 Masukkan ID produk: 009
❌ ID tidak ditemukan. Sisa percobaan: 2
🆔 Masukkan ID produk: 006
❌ ID tidak ditemukan. Sisa percobaan: 1
🆔 Masukkan ID produk: 002
❗ Konfirmasi beli 'Paket Internet 10GB' seharga Rp 30000? (y/n): n
⚠️Transaksi dibatalkan.
```

Gambar 3.49 Hasil Terminal

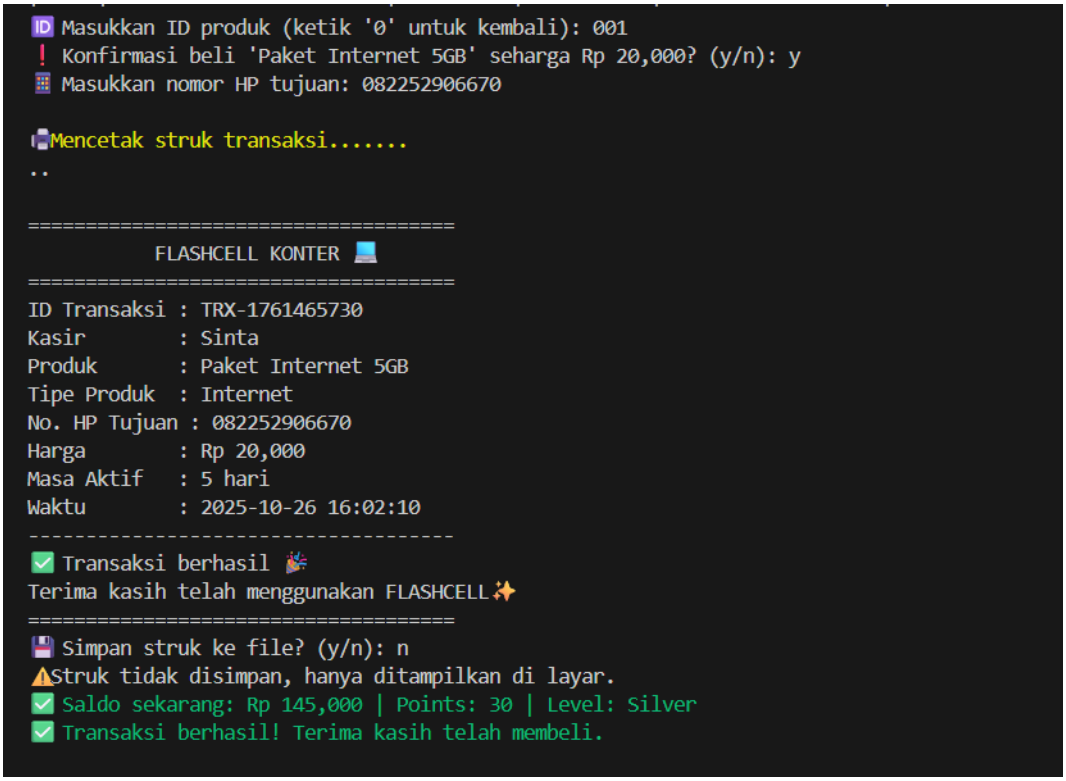
Penjelasan:
Fungsi pilih_id_produk() digunakan untuk memastikan bahwa ID produk yang dimasukkan user valid dan tersedia dalam database (data_produk.csv). Jika pengguna salah mengetik ID, program memberikan tiga kesempatan untuk memperbaiki input. Mekanisme ini mencegah kesalahan transaksi karena salah memilih produk, sehingga lebih aman dan akurat.

7. Cetak Invoice Transaksi

```
def tampilkan_invoice(username, produk_row, harga, masa_aktif, waktu, no_hp="-"):
    cetak_pelan("====================================")
    cetak_pelan("           FLASHCELL KONTER 💻         ")
    cetak_pelan("====================================")
    time.sleep(0.1)
    cetak_pelan(f"ID Transaksi : {invoice_id}")
    cetak_pelan(f"Kasir        : {username}")
    cetak_pelan(f"Produk       : {produk_row.get('nama')}")
    cetak_pelan(f"Tipe Produk  : {produk_row.get('tipe')}")
    cetak_pelan(f"No. HP Tujuan : {no_hp}")
    cetak_pelan(f"Harga        : {format_rp(harga)}")
    cetak_pelan(f"Masa Aktif   : {masa_aktif} hari")
    cetak_pelan(f"Waktu        : {waktu}")
    cetak_pelan("------------------------------------")
    cetak_pelan("✅ Transaksi berhasil 🎉")
    cetak_pelan("Terima kasih telah menggunakan FLASHCELL✨")
    cetak_pelan("====================================")

    # Simpan ke file
    try:
        simpan = input("💾 Simpan struk ke file? (y/n): ").strip().lower()
        if simpan == "y":
            filename = f"data/invoice_{username}_{int(time.time())}.txt"
            with open(filename, "w", encoding="utf-8") as f:
                f.write("====================================\n")
                f.write("           FLASHCELL KONTER 💻         \n")
                f.write("====================================\n")
                f.write(f"ID Transaksi : {invoice_id}\n")
                f.write(f"Kasir        : {username}\n")
                f.write(f"Produk       : {produk_row.get('nama')}\n")
                f.write(f"Tipe Produk  : {produk_row.get('tipe')}\n")
                f.write(f"No. HP Tujuan : {no_hp}\n")
                f.write(f"Harga        : {format_rp(harga)}\n")
                f.write(f"Masa Aktif   : {masa_aktif} hari\n")
                f.write(f"Waktu        : {waktu}\n")
                f.write("------------------------------------\n")
                f.write("✅ Transaksi berhasil 🎉\n")
                f.write("Terima kasih telah menggunakan FLASHCELL✨\n")
                f.write("====================================\n")
            print(f"💾 Struk tersimpan di {filename}")
        else:
            print("⚠️ Struk tidak disimpan, hanya ditampilkan di layar.")
```

Gambar 3.50 Alur Program: Cetak Invoice Transaksi



Gambar 3.51 Hasil Terminal

Penjelasan:
Fungsi tampilkan_invoice() digunakan untuk mencetak struk transaksi setiap kali user berhasil membeli produk. Setiap transaksi otomatis menghasilkan **ID unik (invoice_id)** dengan format waktu (TRX-<timestamp>), sehingga setiap transaksi dapat dibedakan dan dilacak dengan mudah. Struk transaksi juga dapat disimpan sebagai file .txt dalam folder data, menjadikannya lebih aman dan dapat diakses kembali sewaktu-waktu.

8. Error Handling

```python
try:
    while True:
        print("\n=== SISTEM MANAJEMEN PAKET INTERNET DAN PULSA FLASHCELL 💻 ===")
        print("1. Login 🔑")
        print("2. Register ✨")
        print("3. Keluar 🚪")
        pilihan = input("Pilih menu: ").strip()
        if pilihan == "1":
            user = login()
            if user:
                if user.get("role") == "admin":
                    menu_admin()
                else:
                    menu_user(user.get("username"))
        elif pilihan == "2":
            register()
        elif pilihan == "3":
            print("👋 Terima kasih telah berkunjung ke FLASHCELL! Sampai jumpa!")
            break
        else:
            print(Fore.RED + "❌ Pilihan tidak valid!" + Style.RESET_ALL)
except KeyboardInterrupt:
    print("\n❗ Program dihentikan. Sampai jumpa! 👋")
    sys.exit()
```

Gambar 3.52 Alur Program: Error Handling

```
=== SISTEM MANAJEMEN PAKET INTERNET DAN PULSA FLASHCELL 💻 ===
1. Login 🔑
2. Register ✨
3. Keluar 🚪
Pilih menu: q
❌ Pilihan tidak valid!

=== SISTEM MANAJEMEN PAKET INTERNET DAN PULSA FLASHCELL 💻 ===
1. Login 🔑
2. Register ✨
3. Keluar 🚪
Pilih menu:
❗ Program dihentikan. Sampai jumpa! 👋
PS C:\Users\Lenovo\OneDrive\Dokumen\Python>
```

Gambar 3.53 Hasil Terminal

Penjelasan:
Program menggunakan **blok try–except** untuk menangani kemungkinan kesalahan (error) yang terjadi saat runtime.

Beberapa penerapan *error handling* yang terdapat pada program antara lain:

1. ValueError

Digunakan untuk menangkap kesalahan ketika pengguna memasukkan data bukan berupa angka, misalnya pada proses input harga atau saldo top up.

```
try:
    row["harga"] = str(int(harga_input))
except ValueError:
    print(Fore.RED + "❌ Harga harus berupa angka! Tetap menggunakan harga lama." + Style.RESET_ALL)
```

Gambar 3.54 Penerapan: ValueError

2. File Handling Error

Fungsi load_csv() menggunakan *try–except* untuk memastikan bahwa jika file belum ada, program tidak akan berhenti secara mendadak:

```
def load_csv(filename):
    try:
        with open(filename, "r", newline="", encoding="utf-8") as f:
            return list(csv.DictReader(f))
    except Exception:
        return []
```

Gambar 3.55 Penerapan: File Handling Error

3. KeyboardInterrupt & EOFError

Apabila pengguna menekan kombinasi *Ctrl + C*, program akan berhenti dengan pesan yang sopan tanpa menyebabkan error di terminal:

```
try:
    row["harga"] = str(int(harga_input))
except ValueError:
    print(Fore.RED + "❌ Harga harus berupa angka! Tetap menggunakan harga lama." + Style.RESET_ALL)
```

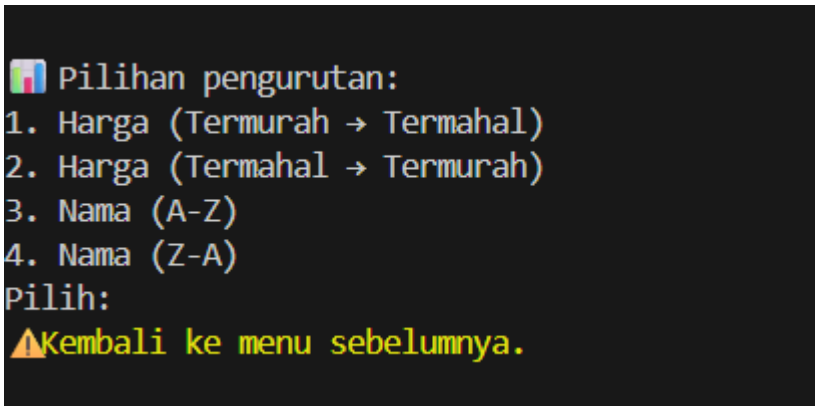
Gambar 3.56 Penerapan: KeyboardInterrupt

```
📝 Masukkan nama produk (0 batal): Paket Internet 11GB
📦 Masukkan tipe (Internet/Pulsa): Internet
💰 Masukkan harga (angka): verv
❌ Harga harus berupa angka!
💰 Masukkan harga (angka):
```

Gambar 3. 57 Hasil Terminal

```
except (KeyboardInterrupt, EOFError):
    print(Fore.YELLOW + "\n⚠️ Kembali ke menu sebelumnya." + Style.RESET_ALL)
    break
```

Gambar 3.58 Penerapan: EOFError



Gambar 3.59 Hasil Terminal

Penjelasan:

ketika pengguna pencet **Ctrl + C** (raise KeyboardInterrupt) atau **Ctrl + Z** (raise EOFError), program **langsung loncat ke except block**,
bukan berhenti atau error — tapi **menampilkan pesan aman**, lalu **break dari loop**, alias kembali ke menu sebelumnya (utama).

9. Reminder Harian

```python
def tampilkan_reminder_harian():
    data_produk = load_csv("data/data_produk.csv")
    daily = [p for p in data_produk if p.get("tipe") == "Internet" and str(p.get("masa_aktif")) == "1"]
    if not daily:
        return

    # Format tanggal bahasa Indonesia
    bulan_list = [
        "Januari", "Februari", "Maret", "April", "Mei", "Juni",
        "Juli", "Agustus", "September", "Oktober", "November", "Desember"
    ]
    hari_list = [
        "Senin", "Selasa", "Rabu", "Kamis",
        "Jumat", "Sabtu", "Minggu"
    ]
    now = datetime.now()
    hari = hari_list[now.weekday()]
    tanggal = f"{hari}, {now.day} {bulan_list[now.month - 1]} {now.year}"

    # Cetak reminder dengan warna
    print(Fore.MAGENTA + "\n🔔 Paket Harian Hari Ini (" + tanggal + "):" + Style.RESET_ALL)
    for p in daily:
        print(Fore.YELLOW + f" • {p.get('nama','').strip()}" + Style.RESET_ALL)
    print()
```

Gambar 3.60 Alur Program: Reminder Harian

```
🔔 Paket Harian Hari Ini (Minggu, 26 Oktober 2025):
 • Paket Unlimited 1 Hari
```

Gambar 3.61 Hasil Terminal

Penjelasan:
Fitur **reminder harian** otomatis menampilkan paket dengan masa aktif 1 hari setiap kali program dijalankan.
Format tanggal ditulis dalam bahasa Indonesia, membuat tampilan lebih natural dan informatif bagi pengguna.

10. Statistik Penjualan Admin

```python
# ----------------------------
# Statistik Penjualan Admin
# ----------------------------
def statistik_admin():
    data_transaksi = load_csv("data/riwayat_transaksi.csv")
    if not data_transaksi:
        print(Fore.YELLOW + "⚠️ Belum ada transaksi, statistik kosong." + Style.RESET_ALL)
        return

    total_transaksi = len(data_transaksi)
    total_pendapatan = sum(int(trx.get("harga",0)) for trx in data_transaksi)

    produk_terjual = {}
    for trx in data_transaksi:
        nama = trx.get("nama_produk","-")
        produk_terjual[nama] = produk_terjual.get(nama,0) + 1

    print(Fore.CYAN + f"\n📊 Statistik Penjualan FLASHCELL" + Style.RESET_ALL)
    print(f"Total Transaksi : {total_transaksi}")
    print(f"Total Pendapatan: {format_rp(total_pendapatan)}")
    print("\nProduk Terjual:")
    for produk, jumlah in produk_terjual.items():
        print(f" • {produk}: {jumlah} kali")
    print()
```

Gambar 3.62 Alur Program : Stastistik Admin

```
======== MENU ADMIN FLASHCELL 👑 ========
1. Lihat Produk 🛍️
2. Tambah Produk ➕
3. Edit Produk 🖍️
4. Hapus Produk 🗑️
5. Lihat Semua Riwayat Transaksi 📜
6. Statistik Penjualan 📈
7. Keluar 🔙
Pilih menu: 6

📊 Statistik Penjualan FLASHCELL
Total Transaksi : 4
Total Pendapatan: Rp 65,000

Produk Terjual:
 • Pulsa 25000: 1 kali
 • Paket Unlimited 1 Hari: 2 kali
 • Paket Internet 5GB: 1 kali
```

Gambar 3.63 Hasil Terminal

Penjelasan:

Fungsi statistik_admin() digunakan untuk **menampilkan rekap penjualan keseluruhan oleh admin**, yang diambil dari file riwayat_transaksi.csv.
Program menghitung jumlah transaksi yang sudah dilakukan (total_transaksi), total

pendapatan yang diperoleh (total_pendapatan), dan menampilkan daftar produk apa saja yang paling sering terjual.

11. Bonus Top Up

```
bonus = int(jumlah*0.05)
total = saldo_lama + jumlah + bonus
user["saldo"] = str(total)
save_csv("data/data_pengguna.csv", users, headers_user)
print(Fore.GREEN + f"✅ Top-up berhasil! Saldo lama: {format_rp(saldo_lama)} → Saldo baru: {format_rp(total)} (bonus {format_rp(bonus)})" + Style.RESET_ALL)
return
```

Gambar 3.64 Alur Program : Bonus Top Up

```
💸 Masukkan jumlah top-up (min 1.000, max 500.000) atau ketik 0 batal: 100000
💰 Memproses top-up...
🏦 Transfer Bank berhasil!
✅ Top-up berhasil! Saldo lama: Rp 115,000 → Saldo baru: Rp 220,000 (bonus Rp 5,000)
```

Gambar 3.65 Hasil Terminal

penjelasan:

Fitur ini merupakan bagian dari **sistem reward otomatis FLASHCELL**, di mana setiap pengguna yang melakukan top-up saldo akan **mendapatkan bonus 5% dari jumlah top-up**.
Contohnya, jika pengguna melakukan top-up sebesar Rp100.000, maka sistem otomatis menambahkan bonus Rp5.000 ke saldo akun.

Fitur ini juga menampilkan simulasi proses pembayaran baik melalui **Bank Transfer** maupun **E-Wallet**, disertai animasi teks pelan menggunakan fungsi cetak_pelan() agar tampilan lebih interaktif.

3.3 Source Code

Tabel 3. 1 Source Code

```
# ----------------------------
# Inisialisasi
# ----------------------------
init(autoreset=True, convert=True)
os.makedirs("data", exist_ok=True)

# ----------------------------
# Header CSV
# ----------------------------
headers_user = ("username", "password", "role",
"saldo", "points", "level")
headers_produk = ("id", "nama", "tipe", "harga",
"masa_aktif")
headers_transaksi = ("username", "nama_produk",
"tipe", "harga", "masa_aktif", "waktu", "no_hp")

# ----------------------------
# Buat file CSV jika belum ada
# ----------------------------
for file, header in [
    ("data/data_pengguna.csv", headers_user),
    ("data/data_produk.csv", headers_produk),
    ("data/riwayat_transaksi.csv", headers_transaksi)
]:
    if not os.path.exists(file):
        with open(file, "w", newline="", encoding="utf-
8") as f:
            writer = csv.writer(f)
            writer.writerow(header)

# ----------------------------
# Fungsi utilitas CSV
# ----------------------------
def load_csv(filepath):
    try:
            with open(filepath, "r", newline="",
encoding="utf-8") as f:
```

```
            return list(csv.DictReader(f))
    except Exception:
        return []

def save_csv(filepath, data, fieldnames):
    with open(filepath, "w", newline="", encoding="utf-
8") as f:
                    writer    =    csv.DictWriter(f,
fieldnames=fieldnames)
        writer.writeheader()
        writer.writerows(data)

def cetak_pelan(teks, delay=0.02):
    for huruf in teks:
        sys.stdout.write(huruf)
        sys.stdout.flush()
        time.sleep(delay)
    print()

def format_rp(angka):
    try:
        return f"Rp {int(angka):,}"
    except Exception:
        return f"Rp {angka}"

def hitung_level(points):
    try:
        p = int(points)
    except Exception:
        p = 0
    if p >= 300:
        return "Platinum"
    elif p >= 100:
        return "Gold"
    else:
        return "Silver"

# -----------------------------
# Data Default
# -----------------------------
def buat_admin_default():
    users = load_csv("data/data_pengguna.csv")
    if not any(u.get("role") == "admin" for u in users):
        users.append({
            "username": "admin",
            "password": "000",
            "role": "admin",
            "saldo": "1000000",
            "points": "0",
            "level": "Platinum"
        })
```

```
            save_csv("data/data_pengguna.csv",  users,
headers_user)

def buat_produk_default():
    data = load_csv("data/data_produk.csv")
    if len(data) == 0:
        produk_awal = [
            {"id": "001", "nama": "Paket Internet 5GB",
"tipe":  "Internet",  "harga":  "20000",  "masa_aktif":
"5"},
          {"id": "002", "nama": "Paket Internet 10GB",
"tipe":  "Internet",  "harga":  "30000",  "masa_aktif":
"10"},
          {"id": "003", "nama": "Pulsa 25000", "tipe":
"Pulsa", "harga": "25000", "masa_aktif": "0"},
          {"id": "004", "nama": "Pulsa 50000", "tipe":
"Pulsa", "harga": "50000", "masa_aktif": "0"},
             {"id": "005", "nama": "Paket Unlimited 1
Hari",   "tipe":   "Internet",   "harga":   "10000",
"masa_aktif": "1"}
        ]
         save_csv("data/data_produk.csv", produk_awal,
headers_produk)

# -----------------------------
# Dashboard User
# -----------------------------
def tampilkan_dashboard(user):
     cetak_pelan(f"\n💎 Dashboard {user['username']}
({user.get('level','Silver')} Member)")
    cetak_pelan(f"Saldo: {format_rp(user.get('saldo',
'0'))}")
    cetak_pelan(f"Points: {user.get('points', '0')}")
    cetak_pelan("🔥 Promo Hari ini: Top-up +5% bonus
saldo!")
    if int(user.get("saldo", 0)) < 50000:
         cetak_pelan(Fore.YELLOW + "⚠ Saldo rendah!
Segera lakukan top-up." + Style.RESET_ALL)

# -----------------------------
# Fungsi Registrasi
# -----------------------------
def register():
    users = load_csv("data/data_pengguna.csv")
    while True:
        try:
            username = input(Fore.CYAN + "✨ Masukkan
username   baru   (3-12)   ketik   0   batal:   "   +
Style.RESET_ALL).strip()
            if username == "0":
```

```
                print(Fore.YELLOW + "⚠ Registrasi
dibatalkan." + Style.RESET_ALL)
                return
            if not (3 <= len(username) <= 12):
                print(Fore.RED + "✖ Username harus 3-
12 karakter!" + Style.RESET_ALL)
                continue
            if any(u["username"] == username for u in
users):
                print(Fore.RED + "✖ Username sudah
terdaftar!" + Style.RESET_ALL)
                continue

            password = pwinput.pwinput(Fore.CYAN + "🔒
Masukkan password baru (4-8) ketik 0 batal: " +
Style.RESET_ALL)
            if password == "0":
                print(Fore.YELLOW + "⚠ Registrasi
dibatalkan." + Style.RESET_ALL)
                return
            if not (4 <= len(password) <= 8):
                print(Fore.RED + "✖ Password harus 4-
8 karakter!" + Style.RESET_ALL)
                continue

            users.append({
                "username": username,
                "password": password,
                "role": "user",
                "saldo": "200000",
                "points": "0",
                "level": "Silver"
            })
            save_csv("data/data_pengguna.csv", users,
headers_user)
            print(Fore.GREEN + f"✅ Registrasi berhasil!
Saldo awal: {format_rp(200000)} 🎉" + Style.RESET_ALL)
            return
        except (KeyboardInterrupt, EOFError):
                print(Fore.YELLOW + "\n⚠ Registrasi
dibatalkan." + Style.RESET_ALL)
            return

# ----------------------------
# Fungsi Login
# ----------------------------
def login():
    users = load_csv("data/data_pengguna.csv")
    try:
```

```
            username = input(Fore.CYAN + "👤 Masukkan
username: " + Style.RESET_ALL).strip()
            password = pwinput.pwinput(Fore.CYAN + "🔑
Masukkan password: " + Style.RESET_ALL)
        for user in users:
                if user["username"] == username and
user["password"] == password:
                                        user["level"]   =
hitung_level(user.get("points", "0"))
                    print(Fore.GREEN + f"\n🎉 Selamat
datang, {username}! Role: {user['role']}. Saldo:
{format_rp(user['saldo'])}\n" + Style.RESET_ALL)
                if user["role"] == "user":
                    tampilkan_dashboard(user)
                return user
        print(Fore.RED + "❌ Login gagal! Username atau
password salah." + Style.RESET_ALL)
        return None
    except (KeyboardInterrupt, EOFError):
        print(Fore.YELLOW + "\n⚠ Login dibatalkan." +
Style.RESET_ALL)
        return None

# ----------------------------
# Top-up Saldo (Bank/E-Wallet)
# ----------------------------
def top_up_saldo(username):
    users = load_csv("data/data_pengguna.csv")
        user = next((u for u in users if
u["username"]==username), None)
    if not user:
        print(Fore.RED + "❌ Akun tidak ditemukan!" +
Style.RESET_ALL)
        return

    percobaan = 0
    saldo_lama = int(user.get("saldo",0))
    while percobaan < 3:
        try:
            print("\n💳 Pilih metode top-up:")
            print("1. Bank Transfer")
            print("2. E-Wallet (OVO/Gopay/Dana)")
            metode = input("Pilih: ").strip()
            if metode not in ["1","2"]:
                  print(Fore.RED + "❌ Pilihan tidak
valid!" + Style.RESET_ALL)
                continue
```

```
            inp = input(f"💸 Masukkan jumlah top-up
(min 1.000, max 500.000) atau ketik 0 batal: ").strip()
            if inp == "0":
                    print(Fore.YELLOW + "⚠ Top-up
dibatalkan." + Style.RESET_ALL)
                return
            jumlah = int(inp)
            if jumlah < 1000 or jumlah > 500000:
                  print(Fore.RED + "✖ Jumlah top-up
tidak sesuai!" + Style.RESET_ALL)
                percobaan +=1
                continue

            # Simulasi proses top-up
            cetak_pelan("💰 Memproses top-up...",0.05)
            time.sleep(0.5)
            if metode=="1":
                    cetak_pelan("🏦 Transfer Bank
berhasil!")
            else:
                    cetak_pelan("📱 Top-up E-Wallet
berhasil!")

            bonus = int(jumlah*0.05)
            total = saldo_lama + jumlah + bonus
            user["saldo"] = str(total)
             save_csv("data/data_pengguna.csv", users,
headers_user)
             print(Fore.GREEN + f"✔ Top-up berhasil!
Saldo lama: {format_rp(saldo_lama)} → Saldo baru:
{format_rp(total)} (bonus {format_rp(bonus)})" +
Style.RESET_ALL)
            return
        except ValueError:
            print(Fore.RED + "✖ Harus berupa angka!"
+ Style.RESET_ALL)
            percobaan +=1
        except (KeyboardInterrupt, EOFError):
                  print(Fore.YELLOW + "\n⚠ Top-up
dibatalkan." + Style.RESET_ALL)
            return

# ----------------------------
# Tampilkan Produk
# ----------------------------
def tampilkan_produk(data=None):
    if data is None:
        data = load_csv("data/data_produk.csv")
    if not data:
```

```
        print(Fore.YELLOW + "⚠ Belum ada data produk."
+ Style.RESET_ALL)
        return
    tabel = PrettyTable()
    tabel.field_names = ["ID", "Nama Produk", "Tipe",
"Harga", "Masa Aktif (hari)"]
    for p in data:
        masa = p.get("masa_aktif")
        try:
            masa_int = int(masa)
            if p.get("tipe")=="Internet" and masa_int
< 1:
                masa_int = 1
            elif p.get("tipe")=="Pulsa":
                masa_int = 0
        except (TypeError, ValueError):
            masa_int = 1 if p.get("tipe")=="Internet"
else 0
        tabel.add_row([p.get("id"), p.get("nama"),
p.get("tipe"), format_rp(p.get("harga")), f"{masa_int}
hari"])
    print(tabel)

# ----------------------------
# Cari Produk Interaktif
# ----------------------------
def cari_produk_interaktif(data):
    if not data:
        print(Fore.YELLOW + "⚠ Belum ada data produk."
+ Style.RESET_ALL)
        return []

    keyword = input("🔍 Masukkan kata kunci
(nama/tipe/harga): ").strip().lower()
    hasil = [
        p for p in data
        if keyword in p.get("nama", "").lower()
        or keyword in p.get("tipe", "").lower()
        or keyword in str(p.get("harga", "")).lower()
    ]

    if hasil:
        print(Fore.GREEN + f"\n✅ Ditemukan
{len(hasil)} produk yang cocok:" + Style.RESET_ALL)
        tampilkan_produk(hasil)
    else:
        print(Fore.YELLOW + "\n⚠ Tidak ada produk yang
cocok." + Style.RESET_ALL)
    return hasil
```

```
# ----------------------------
# Urutkan Produk Interaktif
# ----------------------------
def urutkan_produk_interaktif(data):
    if not data:
        print(Fore.YELLOW + "⚠ Belum ada data produk."
+ Style.RESET_ALL)
        return data

    print("\n📊 Pilihan pengurutan:")
    print("1. Harga (Termurah → Termahal)")
    print("2. Harga (Termahal → Termurah)")
    print("3. Nama (A-Z)")
    print("4. Nama (Z-A)")
    opsi = input("Pilih: ").strip()

    if opsi == "1":
        data_sorted = sorted(data, key=lambda x:
int(x.get("harga", 0)))
    elif opsi == "2":
        data_sorted = sorted(data, key=lambda x:
int(x.get("harga", 0)), reverse=True)
    elif opsi == "3":
        data_sorted = sorted(data, key=lambda x:
x.get("nama", "").lower())
    elif opsi == "4":
        data_sorted = sorted(data, key=lambda x:
x.get("nama", "").lower(), reverse=True)
    else:
        print(Fore.RED + "❌ Pilihan tidak valid!" +
Style.RESET_ALL)
        return data

    print(Fore.GREEN + "\n✅ Data berhasil diurutkan:"
+ Style.RESET_ALL)
    tampilkan_produk(data_sorted)
    return data_sorted

# ---------------------------------------
# Pilih ID Produk (untuk edit/hapus/beli)
# ---------------------------------------
def pilih_id_produk(data, max_attempt=3):
    percobaan = 0
    while percobaan < max_attempt:
        try:
            id_produk = input("🆔 Masukkan ID produk
(ketik '0' untuk kembali): ").strip()
            if id_produk == "0":
```

```
                return None
            for row in data:
                if row.get("id") == id_produk:
                    return row
            percobaan += 1
            print(Fore.RED + f"❌ ID tidak ditemukan.
Sisa     percobaan:     {max_attempt-percobaan}"     +
Style.RESET_ALL)
        except (KeyboardInterrupt, EOFError):
           print(Fore.YELLOW + "\n⚠ Input dibatalkan.
Kembali ke menu sebelumnya." + Style.RESET_ALL)
            return None
     print(Fore.RED + "❌ Terlalu banyak kesalahan.
Kembali ke menu sebelumnya." + Style.RESET_ALL)
    return None

# ----------------------------
# Tambah Produk
# ----------------------------
def tambah_produk():
    data = load_csv("data/data_produk.csv")
    last_id = int(data[-1].get("id")) if data else 0
    id_produk = str(last_id + 1).zfill(3)
    try:
       nama = input("📝 Masukkan nama produk (0 batal):
").strip()
        if nama == "0" or nama == "":
            print(Fore.YELLOW + "⚠ Penambahan produk
dibatalkan." + Style.RESET_ALL)
            return

        while True:
                tipe = input("📄 Masukkan tipe
(Internet/Pulsa): ").strip().title()
            if tipe == "0":
                print(Fore.YELLOW + "⚠ Penambahan
produk dibatalkan." + Style.RESET_ALL)
                return
            if tipe in ["Internet", "Pulsa"]:
                break
            print(Fore.RED + "❌ Tipe produk harus
'Internet' atau 'Pulsa'." + Style.RESET_ALL)

        while True:
            harga_input = input("💰 Masukkan harga
(angka): ").strip()
            if harga_input == "0":
```

```
                    print(Fore.YELLOW + "⚠ Penambahan
produk dibatalkan." + Style.RESET_ALL)
                return
            try:
                harga = int(harga_input)
                break
            except ValueError:
                print(Fore.RED + "✖ Harga harus berupa
angka!" + Style.RESET_ALL)

        # Input masa aktif dengan pernyataan jelas
        while True:
            masa_aktif_input = input("⏳ Masukkan masa
aktif (angka saja, contoh: 20 → otomatis dianggap 20
hari, min 1 untuk Internet, 0 untuk Pulsa): ").strip()
              if masa_aktif_input == "0" and tipe ==
"Pulsa":
                masa_aktif = "0"
                break
                   if masa_aktif_input.isdigit() and
int(masa_aktif_input) > 0:
                masa_aktif = masa_aktif_input
                break
           print(Fore.RED + "✖ Masa aktif harus berupa
angka. Tidak perlu menulis 'hari'!" + Style.RESET_ALL)

          data.append({"id": id_produk, "nama": nama,
"tipe": tipe, "harga": str(harga), "masa_aktif":
masa_aktif})
               save_csv("data/data_produk.csv", data,
headers_produk)
            print(Fore.GREEN + "✔ Produk berhasil
ditambahkan!" + Style.RESET_ALL)
    except (KeyboardInterrupt, EOFError):
          print(Fore.YELLOW + "\n⚠ Penambahan produk
dibatalkan." + Style.RESET_ALL)

# ----------------------------
# Edit Produk
# ----------------------------
def edit_produk():
    data = load_csv("data/data_produk.csv")
    tampilkan_produk(data)
    row = pilih_id_produk(data)
    if not row:
        return
    try:
                new_name = input(f"📝 Nama baru
({row.get('nama')}) [ketik 0 batal]: ").strip()
```

```
        if new_name == "0":
                print(Fore.YELLOW + "⚠ Edit produk
dibatalkan." + Style.RESET_ALL)
            return
        if new_name:
            row["nama"] = new_name

             tipe_input = input(f"📦 Tipe baru
({row.get('tipe')}): ").strip().title()
        if tipe_input == "0":
                print(Fore.YELLOW + "⚠ Edit produk
dibatalkan." + Style.RESET_ALL)
            return
        if tipe_input in ["Internet", "Pulsa"]:
            row["tipe"] = tipe_input

            harga_input = input(f"💰 Harga baru
({row.get('harga')}): ").strip()
        if harga_input == "0":
                print(Fore.YELLOW + "⚠ Edit produk
dibatalkan." + Style.RESET_ALL)
            return
        if harga_input:
            try:
                row["harga"] = str(int(harga_input))
            except ValueError:
               print(Fore.RED + "❌ Harga tidak valid.
Tetap menggunakan harga lama." + Style.RESET_ALL)

        # Input masa aktif dengan pernyataan jelas
        while True:
             masa_input = input(f"⏳ Masa aktif baru
({row.get('masa_aktif')}) (angka saja, contoh: 20 →
otomatis dianggap 20 hari, min 1 untuk Internet, 0 untuk
Pulsa): ").strip()
            if masa_input == "0" and row.get("tipe")
== "Pulsa":
                row["masa_aktif"] = "0"
                break
           if masa_input.isdigit() and int(masa_input)
> 0:
                row["masa_aktif"] = masa_input
                break
          print(Fore.RED + "❌ Masa aktif harus berupa
angka. Tidak perlu menulis 'hari'!" + Style.RESET_ALL)

             save_csv("data/data_produk.csv", data,
headers_produk)
```

```
        print(Fore.GREEN + "✅ Produk berhasil diedit!"
+ Style.RESET_ALL)
    except (KeyboardInterrupt, EOFError):
            print(Fore.YELLOW + "\n⚠ Edit produk
dibatalkan." + Style.RESET_ALL)

# ----------------------------
# Hapus Produk
# ----------------------------
def hapus_produk():
    data = load_csv("data/data_produk.csv")
    tampilkan_produk(data)
    row = pilih_id_produk(data)
    if not row:
        return
    try:
            confirm = input(f"🌡 Yakin hapus produk
'{row.get('nama')}'? (y/n): ").strip().lower()
        if confirm != "y":
            print(Fore.YELLOW + "⚠ Hapus produk
dibatalkan." + Style.RESET_ALL)
            return
        data.remove(row)
        # Update ID setelah hapus
        for idx in range(len(data)):
            data[idx]["id"] = str(idx + 1).zfill(3)
            save_csv("data/data_produk.csv", data,
headers_produk)
        print(Fore.GREEN + "✅ Produk dihapus & ID
diperbarui!" + Style.RESET_ALL)
    except (KeyboardInterrupt, EOFError):
            print(Fore.YELLOW + "\n⚠ Hapus produk
dibatalkan." + Style.RESET_ALL)

# ----------------------------
# Tampilkan Invoice
# ----------------------------
def tampilkan_invoice(username, produk_row, harga,
masa_aktif, waktu, no_hp="-"):
    invoice_id = f"TRX-{int(time.time())}"

    # Minta nomor HP kalau produk Pulsa atau Internet
    if produk_row.get("tipe") in ["Pulsa", "Internet"]:
        no_hp = input("📱 Masukkan nomor HP tujuan:
").strip()
    else:
        no_hp = "-"

    def cetak_pelan(teks, delay=0.02):
```

```
        for huruf in teks:
            sys.stdout.write(huruf)
            sys.stdout.flush()
            time.sleep(delay)
        print()

    # Cetak struk
       print(Fore.YELLOW + "\n🖨️ Mencetak struk
transaksi......." + Style.RESET_ALL)
    time.sleep(0.5)
       for _ in range(2): sys.stdout.write(".");
sys.stdout.flush(); time.sleep(0.3)
    print("\n")

     cetak_pelan("==================================
=")
      cetak_pelan("                  FLASHCELL KONTER
💻          ")
     cetak_pelan("==================================
=")
    time.sleep(0.1)
    cetak_pelan(f"ID Transaksi : {invoice_id}")
    cetak_pelan(f"Kasir        : {username}")
           cetak_pelan(f"Produk                      :
{produk_row.get('nama')}")
               cetak_pelan(f"Tipe      Produk       :
{produk_row.get('tipe')}")
    cetak_pelan(f"No. HP Tujuan : {no_hp}")
    cetak_pelan(f"Harga        : {format_rp(harga)}")
    cetak_pelan(f"Masa Aktif   : {masa_aktif} hari")
    cetak_pelan(f"Waktu        : {waktu}")
    cetak_pelan("----------------------------------
")
    cetak_pelan("✅ Transaksi berhasil 🎉")
       cetak_pelan("Terima kasih telah menggunakan
FLASHCELL✨")
     cetak_pelan("==================================
=")

    # Simpan ke file
    try:
        simpan = input("💾 Simpan struk ke file? (y/n):
").strip().lower()
        if simpan == "y":
                                    filename     =
f"data/invoice_{username}_{int(time.time())}.txt"
            with open(filename, "w", encoding="utf-8")
as f:
```

```
                f.write("============================
=========\n")
                f.write("            FLASHCELL KONTER
💻         \n")
                f.write("============================
=========\n")
                            f.write(f"ID  Transaksi   :
{invoice_id}\n")
            f.write(f"Kasir        : {username}\n")
                        f.write(f"Produk            :
{produk_row.get('nama')}\n")
                            f.write(f"Tipe  Produk    :
{produk_row.get('tipe')}\n")
            f.write(f"No. HP Tujuan : {no_hp}\n")
                        f.write(f"Harga             :
{format_rp(harga)}\n")
             f.write(f"Masa Aktif    : {masa_aktif}
hari\n")
            f.write(f"Waktu         : {waktu}\n")
            f.write("----------------------------
--------\n")
             f.write("✅ Transaksi berhasil 🎉\n")
            f.write("Terima kasih telah menggunakan
FLASHCELL✨\n")
             f.write("============================
=========\n")
           print(f"💾 Struk tersimpan di {filename}")
        else:
              print("⚠ Struk tidak disimpan, hanya
ditampilkan di layar.")
    except (KeyboardInterrupt, EOFError):
        print("\n⚠ Simpan struk dibatalkan.")

# -------------------------------------
# Lihat Semua Riwayat Transaksi (Admin)
# -------------------------------------
def lihat_semua_riwayat():
                          data_transaksi            =
load_csv("data/riwayat_transaksi.csv")
    if not data_transaksi:
         print(Fore.YELLOW + "⚠ Belum ada transaksi
sama sekali.\n" + Style.RESET_ALL)
        return
    tabel = PrettyTable()
                        tabel.field_names           =
["Username","Produk","Tipe","Harga","Masa  Aktif","No
HP","Waktu"]
    for trx in data_transaksi:
        tabel.add_row([
```

```
            trx.get("username"),
            trx.get("nama_produk"),
            trx.get("tipe"),
            format_rp(trx.get("harga")),
            trx.get("masa_aktif"),
            trx.get("no_hp","-"),
            trx.get("waktu")
        ])
    print(tabel)

# ----------------------------
# Statistik Penjualan Admin
# ----------------------------
def statistik_admin():
                            data_transaksi            =
load_csv("data/riwayat_transaksi.csv")
    if not data_transaksi:
         print(Fore.YELLOW + "⚠ Belum ada transaksi,
statistik kosong." + Style.RESET_ALL)
        return

    total_transaksi = len(data_transaksi)
    total_pendapatan = sum(int(trx.get("harga",0)) for
trx in data_transaksi)

    produk_terjual = {}
    for trx in data_transaksi:
        nama = trx.get("nama_produk","-")
                             produk_terjual[nama]      =
produk_terjual.get(nama,0) + 1

      print(Fore.CYAN  +  f"\n📊  Statistik  Penjualan
FLASHCELL" + Style.RESET_ALL)
    print(f"Total Transaksi : {total_transaksi}")
                     print(f"Total         Pendapatan:
{format_rp(total_pendapatan)}")
    print("\nProduk Terjual:")
    for produk, jumlah in produk_terjual.items():
        print(f" • {produk}: {jumlah} kali")
    print()

# ----------------------------
# Lihat Riwayat Transaksi User
# ----------------------------
def lihat_riwayat(username):
                            data_transaksi            =
load_csv("data/riwayat_transaksi.csv")
     user_trx  =  [trx  for  trx  in  data_transaksi  if
trx.get("username")==username]
    if not user_trx:
```

```
        print(Fore.YELLOW + "\n⚠ Belum ada riwayat
transaksi.\n" + Style.RESET_ALL)
        return
    tabel = PrettyTable()
                tabel.field_names       =       ["Nama
Produk","Tipe","Harga","Masa Aktif","Waktu"]
    for trx in user_trx:
        tabel.add_row([
            trx.get("nama_produk"),
            trx.get("tipe"),
            format_rp(trx.get("harga")),
            trx.get("masa_aktif","0"),
            trx.get("waktu")
        ])
    print(tabel)

# ------------------------------------------------
-----------------------------
# Beli Produk + Update Saldo, Points, Level + Undo
(dengan input no HP untuk Pulsa)
# ------------------------------------------------
-----------------------------
def beli_produk(username):
    data_produk = load_csv("data/data_produk.csv")
    tampilkan_produk(data_produk)
    produk_row = pilih_id_produk(data_produk)
    if not produk_row:
        return
    users = load_csv("data/data_pengguna.csv")
        user   =   next((u   for   u   in   users   if
u["username"]==username), None)
    if not user:
        print(Fore.RED + "✖ Akun tidak ditemukan!" +
Style.RESET_ALL)
        return
    harga = int(produk_row.get("harga",0))
    masa_aktif = produk_row.get("masa_aktif","0")
    saldo_user = int(user.get("saldo",0))
    if saldo_user < harga:
         print(Fore.RED + "✖ Saldo tidak cukup!" +
Style.RESET_ALL)
        return

    # Input nomor HP untuk Pulsa
    no_hp = ""
    if produk_row.get("tipe") == "Pulsa":
        while True:
            no_hp = input("📱 Masukkan nomor HP tujuan
(9-15 digit): ").strip()
```

```
            if no_hp.isdigit() and 9 <= len(no_hp) <=
15:
                break
            print(Fore.RED + "❌ Nomor HP harus berupa
angka dan 9-15 digit!" + Style.RESET_ALL)

    try:
            confirm = input(f"❗ Konfirmasi beli
'{produk_row['nama']}' seharga {format_rp(harga)}?
(y/n): ").strip().lower()
        if confirm != "y":
                print(Fore.YELLOW + "⚠ Transaksi
dibatalkan." + Style.RESET_ALL)
            return
    except (KeyboardInterrupt, EOFError):
            print(Fore.YELLOW + "\n⚠ Transaksi
dibatalkan." + Style.RESET_ALL)
        return

        waktu = datetime.now().strftime("%Y-%m-%d
%H:%M:%S")
    transaksi = {
        "username": username,
        "nama_produk": produk_row.get("nama"),
        "tipe": produk_row.get("tipe"),
        "harga": str(harga),
        "masa_aktif": masa_aktif,
        "waktu": waktu,
        "no_hp": no_hp if no_hp else "-"
    }

    # Update saldo, points, level
    user["saldo"] = str(saldo_user - harga)
    user["points"] = str(int(user.get("points",0))+10)
    user["level"] = hitung_level(user.get("points",0))
        save_csv("data/data_pengguna.csv", users,
headers_user)

    # Simpan transaksi
                        data_transaksi =
load_csv("data/riwayat_transaksi.csv")
    data_transaksi.append(transaksi)
            save_csv("data/riwayat_transaksi.csv",
data_transaksi, headers_transaksi)

    # Tampilkan invoice
      tampilkan_invoice(username, produk_row, harga,
masa_aktif, waktu)
```

```
        print(Fore.GREEN + f"✅ Saldo sekarang: {format_rp(user['saldo'])} | Points: {user['points']} | Level: {user['level']}" + Style.RESET_ALL)
    print(Fore.GREEN + "✅ Transaksi berhasil! Terima kasih telah membeli." + Style.RESET_ALL)

#-------------------
# Reminder harian
#-------------------
def tampilkan_reminder_harian():
    data_produk = load_csv("data/data_produk.csv")
    daily = [p for p in data_produk if p.get("tipe") == "Internet" and str(p.get("masa_aktif")) == "1"]
    if not daily:
        return

    # Format tanggal bahasa Indonesia
    bulan_list = [
        "Januari", "Februari", "Maret", "April", "Mei", "Juni",
        "Juli", "Agustus", "September", "Oktober", "November", "Desember"
    ]
    hari_list = [
        "Senin", "Selasa", "Rabu", "Kamis",
        "Jumat", "Sabtu", "Minggu"
    ]
    now = datetime.now()
    hari = hari_list[now.weekday()]
    tanggal = f"{hari}, {now.day} {bulan_list[now.month - 1]} {now.year}"

    # Cetak reminder dengan warna
    print(Fore.MAGENTA + "\n🔔 Paket Harian Hari Ini (" + tanggal + "):" + Style.RESET_ALL)
    for p in daily:
        print(Fore.YELLOW + f"  • {p.get('nama','').strip()}" + Style.RESET_ALL)
    print()

# ----------------------------
# Menu Utama Admin
# ----------------------------
def menu_admin():
    while True:
        try:
            print("\n======== MENU ADMIN FLASHCELL 🐱 ========")
            print("1. Lihat Produk 🛍")
```

```
            print("2. Tambah Produk ➕")
            print("3. Edit Produk ✏️")
            print("4. Hapus Produk 🗑️")
              print("5. Lihat Semua Riwayat Transaksi
📜")
            print("6. Statistik Penjualan 📈")
            print("7. Keluar 🔙")
            pilihan = input("Pilih menu: ").strip()
            if pilihan=="1":
                menu_lihat_produk()
            elif pilihan=="2":
                tambah_produk()
            elif pilihan=="3":
                edit_produk()
            elif pilihan=="4":
                hapus_produk()
            elif pilihan=="5":
                lihat_semua_riwayat()
            elif pilihan=="6":
                statistik_admin()
            elif pilihan=="7":
                break
            else:
                  print(Fore.RED + "❌ Pilihan tidak
valid!" + Style.RESET_ALL)
        except (KeyboardInterrupt, EOFError):
            print(Fore.YELLOW + "\n⚠️ Kembali ke menu
utama admin." + Style.RESET_ALL)
            break

# ----------------------------
# Menu Utama User
# ----------------------------
def menu_user(username):
    while True:
        try:
             print(f"\n======== MENU USER ({username})
👤 ========")
            print("1. Lihat Produk 🏭")
            print("2. Beli Produk 🛒")
            print("3. Top Up Saldo 💸")
            print("4. Lihat Riwayat Transaksi 📜")
            print("5. Keluar 🔙")
            pilihan = input("Pilih menu: ").strip()
            if pilihan=="1":
                menu_lihat_produk()
            elif pilihan=="2":
```

```
                beli_produk(username)
            elif pilihan=="3":
                top_up_saldo(username)
            elif pilihan=="4":
                lihat_riwayat(username)
            elif pilihan=="5":
                break
            else:
                  print(Fore.RED + "❌ Pilihan tidak
valid!" + Style.RESET_ALL)
        except (KeyboardInterrupt, EOFError):
            print(Fore.YELLOW + "\n⚠ Kembali ke menu
utama user." + Style.RESET_ALL)
            break

# ----------------------------
# Menu Lihat Produk Interaktif
# ----------------------------
def menu_lihat_produk():
    data = load_csv("data/data_produk.csv")
    while True:
        try:
            print("\n---- MENU LIHAT PRODUK ----")
            print("1. Tampilkan semua produk 🏪")
            print("2. Cari produk 🔍")
            print("3. Urutkan produk 📊")
            print("4. Kembali 🔙")
            pilihan = input("Pilih: ").strip()

            if pilihan == "1":
                tampilkan_produk(data)
            elif pilihan == "2":
                cari_produk_interaktif(data)
            elif pilihan == "3":
                urutkan_produk_interaktif(data)
            elif pilihan == "4":
                break
            else:
                  print(Fore.RED + "❌ Pilihan tidak
valid!" + Style.RESET_ALL)

        except (KeyboardInterrupt, EOFError):
            print(Fore.YELLOW + "\n⚠ Kembali ke menu
sebelumnya." + Style.RESET_ALL)
            break

# ----------------------------
# Program Utama
# ----------------------------
```

```
def main():
    buat_admin_default()
    buat_produk_default()

    tampilkan_reminder_harian()

    while True:
        try:
           print("\n=== SISTEM MANAJEMEN PAKET INTERNET
DAN PULSA FLASHCELL 💻 ===")
            print("1. Login 🔑")
            print("2. Register 🌟")
            print("3. Keluar 🚪")
            pilihan = input("Pilih menu: ").strip()
            if pilihan=="1":
                user = login()
                if user:
                    if user["role"]=="admin":
                        menu_admin()
                    else:
                        menu_user(user["username"])
            elif pilihan=="2":
                register()
            elif pilihan=="3":
               print(Fore.CYAN + "\nTerima kasih telah
menggunakan FLASHCELL 💔" + Style.RESET_ALL)
                break
            else:
                  print(Fore.RED + "❌ Pilihan tidak
valid!" + Style.RESET_ALL)
        except (KeyboardInterrupt, EOFError):
           print(Fore.YELLOW + "\n⚠ Program dihentikan
oleh pengguna." + Style.RESET_ALL)
            break

# ---------------------------
# Jalankan Program
# ---------------------------
if __name__=="__main__":
    main()
```

BAB IV
PENUTUP

4.1 Kesimpulan

Program **Sistem Manajemen Paket Internet & Pulsa – FLASHCELL** berhasil dibuat menggunakan bahasa pemrograman **Python** dengan mengimplementasikan berbagai modul dan konsep yang telah dipelajari selama praktikum, seperti CSV, pwinput, PrettyTable, os, datetime, serta pengelolaan data menggunakan *dictionary*.

Program ini mampu mengelola data pengguna dan produk secara dinamis, menyediakan fitur **CRUD (Create, Read, Update, Delete)** untuk admin, serta memberikan kemudahan bagi pengguna dalam melakukan **pembelian paket internet atau pulsa**, **top-up saldo dengan bonus otomatis**, dan **melihat riwayat transaksi**.

Selain itu, sistem dilengkapi dengan berbagai fitur pendukung seperti:

- Reminder harian otomatis untuk paket tertentu.
- Pencarian dan pengurutan produk (search & sort) agar data mudah diakses.
- Statistik penjualan untuk admin sebagai laporan hasil transaksi.
- Error handling yang menjaga kestabilan sistem ketika terjadi kesalahan input atau interupsi pengguna.
- Cetak invoice digital yang menyertakan data lengkap transaksi pelanggan.

Dengan demikian, aplikasi ini telah memenuhi tujuan pembuatan sistem, yaitu sebagai **media simulasi manajemen transaksi penjualan pulsa dan paket internet secara digital, cepat, dan terstruktur**.

4.2 Saran

Agar sistem FLASHCELL dapat berkembang lebih baik ke depannya, berikut beberapa saran pengembangan:

1. Menambahkan **fitur login menggunakan database SQL** agar keamanan dan skalabilitas data lebih terjamin.
2. Mengembangkan tampilan program dengan **Graphical User Interface (GUI)** menggunakan tkinter atau PyQt agar lebih interaktif.
3. Menambahkan sistem **notifikasi otomatis berbasis email atau WhatsApp** untuk konfirmasi transaksi.

4. Mengimplementasikan **fitur laporan bulanan otomatis** untuk admin agar proses monitoring penjualan lebih mudah.
5. Menambahkan **fitur API integrasi operator** untuk simulasi pembelian pulsa yang lebih realistis.

DAFTAR PUSTAKA

Wahyudi, T., & Rose Handayani, V. (2025). Perancangan Sistem Informasi Penjualan Paket Internet Berbasis Website Untuk Peningkatan Layanan Pada PT Telekomunikasi Seluler Purwokerto. *Jurnal Sains Dan Manajemen*, *13*(1).

LAMPIRAN

Lampiran 1 : Tabel Kontribusi

Nama	Kontribusi	Bagian
Siti Nursinta (2509116087)	Konsep, Coding, Laporan, Pengecekan, Pengembangan	1. konsep program 2. Coding 3. Menjelaskan program di Laporan 4. Mencari bug dan *error* 5. Konsep dan penyusunan Laporan 6. Pengembangan program
Muhammad Rizky Alfa Reza Basyah (2509116086)	Flowchart, Readme, Laporan	1. Flowchart program 2. Bagian Laporan Flowchart 3. Menulis Readme 4. Pengecekan laporan dan Program
Adelia Githa Naveezha Hermawan (2509116110)	pengecekan	1. Pengecekan Flowchart dan Readme

Gambar 1.1 Lampiran: Konsul Ke 1

Gambar 1.2 Lampiran: Konsul Ke 2