**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

[Kata Pengantar ii](#_Toc148437917)

[Daftar Isi iii](#_Toc148437918)

[Daftar Gambar iii](#_Toc148437919)

[Daftar Tabel vii](#_Toc148437920)

[BAB I 8](#_Toc148437921)

[1.1 Deskripsi Masalah 8](#_Toc148437922)

[1.2 Rumusan Masalah 8](#_Toc148437923)

[1.3 Batasan Masalah 8](#_Toc148437924)

[1.4 Tujuan 8](#_Toc148437925)

[1.5 Manfaat 8](#_Toc148437926)

[BAB II 9](#_Toc148437927)

[2.1 Analisis Program 9](#_Toc148437928)

[2.2 Flowchart 9](#_Toc148437929)

[BAB III 10](#_Toc148437930)

[3.1 Implementasi Program 10](#_Toc148437931)

[3.2 Alur Program 10](#_Toc148437932)

[3.3 Source Code 10](#_Toc148437933)

[BAB IV 11](#_Toc148437934)

[4.1 Kesimpulan 12](#_Toc148437935)

[4.2 Saran 12](#_Toc148437936)

[Daftar Pustaka 13](#_Toc148437937)

[Lampiran 14](#_Toc148437938)

# DAFTAR GAMBAR

**Gambar 2.1** Flowchart menu utama ............................................................. 14

**Gambar 2.2** Flowchart menu admin.............................................................. 15

**Gambar 2.3** Flowchart 2.3 ........................................................................... 16

**Gambar 2.4** Flowchart 2.4 ........................................................................... 17

**Gambar 2.5** Flowchart 2.5 ........................................................................... 18

**Gambar 2.6** Flowchart 2.6 ........................................................................... 19

**Gambar 2.7** Flowchart 2.7 ........................................................................... 16

**Gambar 2.8** Flowchart 2.8 ........................................................................... 17

**Gambar 2.9** Flowchart 2.9 ........................................................................... 18

**Gambar 2.10** Flowchart 2.10 ....................................................................... 19

**Gambar 3.11** Flowchart 2.11 ……............................................................... 20

**Gambar 3.12** Flowchart 2.12 ……............................................................... 20

**Gambar 3.1** Implementasi: Python .............................................................. 20

**Gambar 3.2** Implementasi: Module …………………………..................... 21

**Gambar 3.3** Implementasi: CSV………………………………………...….21

**Gambar 3.4** Implementasi: Def (Funcation)……………............................. 21

**Gambar 3.5** Implementasi: If Elif Else…………………............................. 21

**Gambar 3.6** Implementasi: While…………………………......................... 21

**Gambar 3.7** Implementasi: CRUD…………………………………………22

**Gambar 3.8** Implementasi: Create …………………………………........... 22

**Gambar 3.9** Implementasi: Read ………………………............................. 23

**Gambar 3.10** Implementasi: Update ……………........................................ 24

**Gambar 3.11** Implementasi: Delete ………………..................................... 24

**Gambar 3.12** Implementasi: ValueError ………………............................. 25

**Gambar 3.13** Implementasi: KeyboardInterruot ......................................... 25

**Gambar 3.14** Alur Program: Menu Utama ................................................. 26

**Gambar 3.15** Alur Program: Hasil Terminal .............................................. 26

**Gambar 3.16** Alur Program: Registrasi ...................................................... 26

**Gambar 3.17** Alur Program: Hasil Terminal .............................................. 27

**Gambar 3.18** Alur Program: Login Sistem …............................................. 27

**Gambar 3.19** Alur Program: Login: Admin ............................................... 28

**Gambar 3.20** Alur Program: Menu Admin …………….............................. 28

**Gambar 3.21** Alur Program: Hasil Terminal ………………….................. 28

**Gambar 3.22** Alur Program: Admin-lihat produk ....................................... 29

**Gambar 3.23** Alur Program: Hasil Teminal ……………………................ 29

**Gambar 3.24** Alur Program: Admin-Tambah Produk ................................. 29

**Gambar 3.25** Alur Program: Hasil Terminsl ………………………………29

**Gambar 3.26** Alur Program: Admin-Edit Produk ……………................... 30

**Gambar 3.27** Alur Program: Hasil Terminal ………………………………30

**Gambar 3.28** Alur Program: Admin-Hapus Produk …………………….…30

**Gambar 3.29** Alur Program: Hasil Terminal ……………….……….……..30

**Gambar 3.30** Alur Program: Admin-Lihat Semua Transaksi....................... 31

**Gambar 3.31** Alur Program: Hasil Terminal ............................................... 31

**Gambar 3.32** Alur Program: Admin-Logout ……....................................... 32

**Gambar 3.33** Alur Program: Admin-Cari Produk........................................ 32

**Gambar 3.34** Alur Program: Hasil Terminal ………................................... 33

**Gambar 3.35** Alur Program: Admin-Urutkan Produk.................................. 34

**Gambar 3.36** Tampilan Program: Hasil Terminal ....................................... 53

**Gambar 3.37** Tampilan Program: Menu User ............................................. 54

**Gambar 3.38** Tampilan Program: Hasil Terminal ....................................... 54

**Gambar 3.39** Tampilan Program: User-Lihat Produk …............................. 54

**Gambar 3.40** Tampilan Program: Hasil Terminal ....................................... 55

**Gambar 3.41** Tampilan Program: User-Beli Produk ................................... 55

**Gambar 3.42** Tampilan Program: Hasil Terminal ....................................... 55

**Gambar 3.43** Tampilan Program: User-Top Up Saldo …............................ 56

**Gambar 3.44** Tampilan Program: Hasil Terminal ……............................... 57

**Gambar 3.45** Tampilan Program: User-Lihat Riwayat Transaksi ............... 58

**Gambar 3.46** Tampilan Program: Hasil Terminal .….................................. 59

**Gambar 3.47** Tampilan Program: User-Logout …………........................... 60

**Gambar 3.48** Tampilan Program: Pemilihan ID Produk …......................... 60

**Gambar 3.49** Tampilan Program: Hasil Terminal ……............................... 61

**Gambar 3.50** Tampilan Program: Cetak Invoice Transaksi ........................ 62

**Gambar 3.51** Tampilan Program: Hasil Terminal ……............................... 63

**Gambar 3.52** Tampilan Program: Error Handling ....................................... 64

**Gambar 3.53** Tampilan Program: Hasil Terminal .…….............................. 64

**Gambar 3.54** Tampilan Program: Penerapan: ValueError .......................... 64

**Gambar 3.55** Tampilan Program: Penerapan: File Handling Error ............ 65

**Gambar 3.56** Tampilan Program: Penerapan KeyboardInterrupt................. 65

**Gambar 3.57** Tampilan Program: Hasil Terminal ....................................... 65

**Gambar 3.58** Tampilan Program: Penerapan EOFerror............................... 66

**Gambar 3.59** Tampilan Program: Hasil Terminal …................................... 67

**Gambar 3.60** Tampilan Program: Reminder Harian ……........................... 67

**Gambar 3.61** Tampilan Program: Hasil Terminal ...................................... 68

**Gambar 3.62** Tampilan Program: Stastistik Admin …................................ 69

**Gambar 3.63** Tampilan Program: Hasil Terminal ………………............... 69

**Gambar 3.64** Tampilan Program: Bonus Top up ..........................................70

**Gambar 3.65** Tampilan Program: Hasil Terminal ........................................ 70

# DAFTAR TABEL

[**Tabel 3. 1** Source Code 3](#_Toc210894655)

# DAFTAR LAMPIRAN

[**Lampiran 1 :** Tabel Kontribusi 6](#_Toc210894679)

# 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:

## Bagaimana merancang Sistem Manajemen Paket Internet dan Pulsa yang efisien dan mudah digunakan?

## Bagaimana sistem dapat dijalankan oleh Admin dengan akses penuh terhadap data produk dan transaksi?

## Bagaimana sistem dapat dijalankan oleh User agar dapat melihat dan membeli produk dengan mudah?

## Bagaimana sistem dapat membantu pencatatan dan pelaporan transaksi secara otomatis dan akurat?

## 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:

## Sistem hanya berfokus pada pengelolaan data produk berupa paket internet dan pulsa.

## Sistem memiliki dua role utama, yaitu Admin dan User.

## Sistem tidak terhubung langsung dengan server operator seluler, melainkan hanya berupa simulasi program.

## Sistem dibuat menggunakan bahasa pemrograman Python dan dijalankan melalui console atau terminal.

## Sistem bersifat sederhana dan digunakan sebagai media pembelajaran.

## 1.4 Tujuan

## Tujuan dari pembuatan dan penyusunan sistem ini adalah:

## Membuat sistem yang dapat membantu pengelolaan data produk, pelanggan, dan transaksi secara digital.

## Mempermudah pengguna dalam melihat dan membeli paket internet serta pulsa.

## Memudahkan admin dalam menambah, memperbarui, dan menghapus data produk serta memantau laporan penjualan.

## Mengimplementasikan konsep CRUD dan sistem multi-role (Admin dan User) menggunakan Python.

## 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. Datetime

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

1. 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.

## A diagram of a flowchart AI-generated content may be incorrect.2.2 Flowchart

**Gambar 2.1** Flowchart Menu Admin

A screenshot of a computer screen

AI-generated content may be incorrect.

**Gambar 2.2** Flowchart Menu Admin

A diagram of a flowchart

AI-generated content may be incorrect.

**Gambar 2.3** Flowchart lihat produk admin

A black background with white text

AI-generated content may be incorrect.

**Gambar 2.4** Flowchart tambah produk admin

A diagram of a cylinder

AI-generated content may be incorrect.A diagram of a product

AI-generated content may be incorrect.

**Gambar 2.7** Flowchart riwayat transaksi admin

**Gambar 2.5** Flowchart edit produk admin



**Gambar 2.6** Flowchart hapus produk admin

A diagram of a diagram

AI-generated content may be incorrect.

**Gambar 2.8** Flowchart menu user

A diagram of a flowchart

AI-generated content may be incorrect.

**Gambar 2.9** Flowchart lihat produk user

A diagram of a diagram

AI-generated content may be incorrect.

**Gambar 2.10** Flowchart beli produk user

A black screen with white text

AI-generated content may be incorrect.

**Gambar 2.11** Flowchart top up saldo

A diagram of a flowchart

AI-generated content may be incorrect.

**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.

1. Dictionary

Struktur data **dictionary** digunakan dalam program ini untuk menyimpan data pengguna, data produk, dan data transaksi. Dictionary dipilih karena mampu menyimpan data dalam bentuk pasangan *key-value*, sehingga A black background with text

AI-generated content may be incorrect.mempermudah pengambilan dan pengelolaan informasi.

**Gambar 3.1** Implementasi : Dictionary

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

1. 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.

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

AI-generated content may be incorrect.

**Gambar 3.2** Implementasi : Import Module

1. CSV (Comma Separated Values)

A computer code on a black background

AI-generated content may be incorrect.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.

**Gambar 3.3** Implementasi : CSV

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

1. 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

A computer screen shot of text

AI-generated content may be incorrect.Penjelasan:

**Gambar 3.4** Implementasi : Def

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.

1. If, Elif, Else

A screen shot of a computer

AI-generated content may be incorrect. Struktur kendali **if, elif, else** digunakan untuk melakukan percabangan logika dalam program. Contohnya pada saat login, sistem akan memverifikasi username dan password.

**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.

1. While

A screen shot of a computer screen

AI-generated content may be incorrect.Struktur perulangan **while** digunakan untuk menampilkan menu utama secara terus-menerus selama pengguna belum memilih keluar dari sistem.

**Gambar 3.6** Implementasi : While

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

1. CRUD (Create, Read, Update, Delete) If, Elif, Else

Program FLASHCELL juga mengimplementasikan operasi CRUD dalam pengelolaan data produk.

A screen shot of a computer program

AI-generated content may be incorrect.a. Create

**Gambar 3.7** Implementasi : Create

A screen shot of a computer program

AI-generated content may be incorrect.b. Read

**Gambar 3.8** Implementasi : Read

c. Update

A computer screen shot of text

AI-generated content may be incorrect.

**Gambar 3.9** Implementasi : Update

A screen shot of a computer code

AI-generated content may be incorrect.d. Delete

**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.

1. 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.

A screen shot of a computer screen

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

**Gambar 3.12** Implementasi : KeyboardInterrupt

**Gambar 3.11** Implementasi : ValueError

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

A black screen with white text

AI-generated content may be incorrect.A screen shot of a computer screen

AI-generated content may be incorrect.

**Gambar 3.14** Hasil Terminal

**Gambar 3.13** Alur Program: Menu Utama

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.

A screen shot of a computer code

AI-generated content may be incorrect.2. Registrasi Pengguna Baru

**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.

**Gambar 3.16** Hasil Terminal

A computer code on a black background

AI-generated content may be incorrect.3. Login ke Sistem

**Gambar 3.17** Alur Program: Login ke Sistem

A screen shot of a computer

AI-generated content may be incorrect.

**Gambar 3.18** Hasil Terminal Admin

**Gambar 3.19** Hasil Terminal User

A screen shot of a computer

AI-generated content may be incorrect.

**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

A screenshot of a black screen

AI-generated content may be incorrect.A screen shot of a computer screen

AI-generated content may be incorrect.

**Gambar 3.21** Hasil Terminal

**Gambar 3.20** Alur Program: Menu Admin

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

A screenshot of a computer

AI-generated content may be incorrect.A screen shot of a computer screen

AI-generated content may be incorrect.

**Gambar 3.23** Hasil Terminal

**Gambar 3.22** Alur Program: Admin – Lihat Produk

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.

A screen shot of a computer

AI-generated content may be incorrect.A screen shot of a computer screen

AI-generated content may be incorrect.b. Tambah Produk

**Gambar 3.25** Hasil Terminal

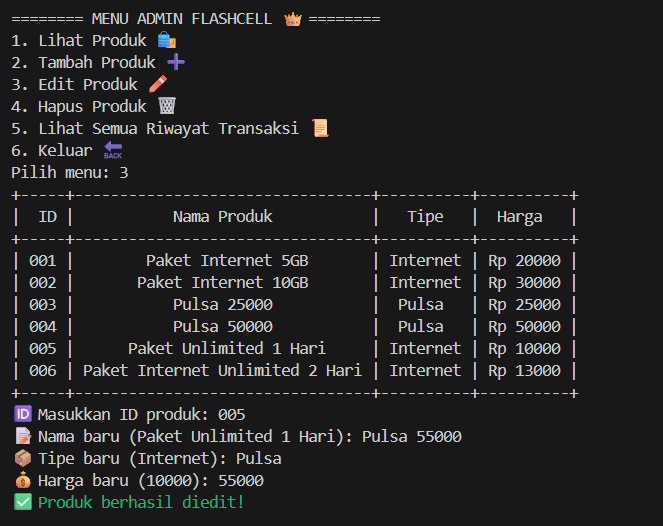
**Gambar 3.24** Alur Program: Admin – Tambah Produk

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

A screenshot of a computer code

AI-generated content may be incorrect.



**Gambar 3.26** Alur Program: Admin – Edit Produk

**Gambar 3.27** Hasil Terminal

Penjelasan:

Fitur **Edit Produk** memungkinkan admin memperbarui data produk yang sudah

ada, seperti nama, tipe, dan harga.

d. Delete Produk

**Gambar 3.29** Hasil Terminal

A screenshot of a computer

AI-generated content may be incorrect.A screen shot of a computer code

AI-generated content may be incorrect.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.

**Gambar 3.28** Alur Program: Admin – Hapus Produk

A black screen with text

AI-generated content may be incorrect.e. Lihat Semua Riwayat Transaksi

**Gambar 3.30** Alur Program: Admin – Riwayat Transaksi

A screenshot of a computer

AI-generated content may be incorrect.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.

**Gambar 3.31** Hasil Terminal

A screenshot of a computer

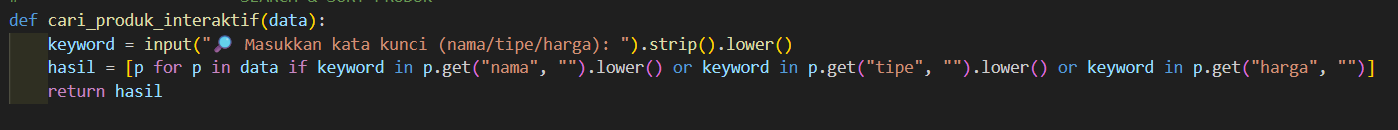
AI-generated content may be incorrect.f. Logout

**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

A screenshot of a computer

AI-generated content may be incorrect.

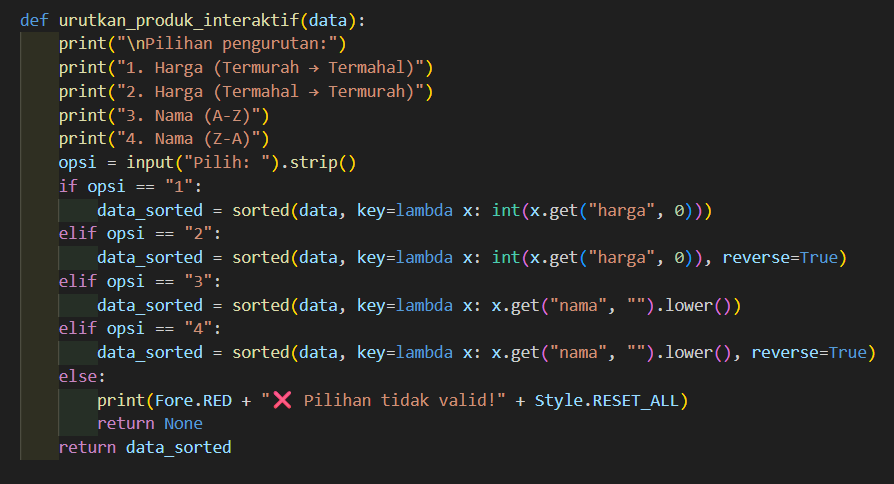
**Gambar 3.34** Hasil Terminal

**Gambar 3.33** Alur Program: Admin – Searching Produk

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

Penjelasan:  
Fitur *sorting* digunakan untuk mengurutkan produk berdasarkan harga atau nama.A screenshot of a computer

AI-generated content may be incorrect. 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.

**Gambar 3.36** Alur Program: Hasil Terminal

**Gambar 3.35** Hasil Terminal

**Gambar 3.35** Program: Admin – Sorting Produk

A screenshot of a computer

AI-generated content may be incorrect.5. Menu User

**Gambar 3.38** Hasil Terminal

**Gambar 3.37** Alur Program: Menu User

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

A screenshot of a computer

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.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.

**Gambar 3.40** Hasil Terminal

**Gambar 3.39** Alur Program: User – Lihat Produk

A screen shot of a computer program

AI-generated content may be incorrect.b. Beli Produk

**Gambar 3.41** Alur Program: User – Beli Produk

Penjelasan:  
Fitur ini digunakan oleh user untuk membeli produk yang tersedia.A screenshot of a computer

AI-generated content may be incorrect. 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.

**Gambar 3.42** Hasil Terminal

A screen shot of a computer program

AI-generated content may be incorrect.c. Top Up Saldo

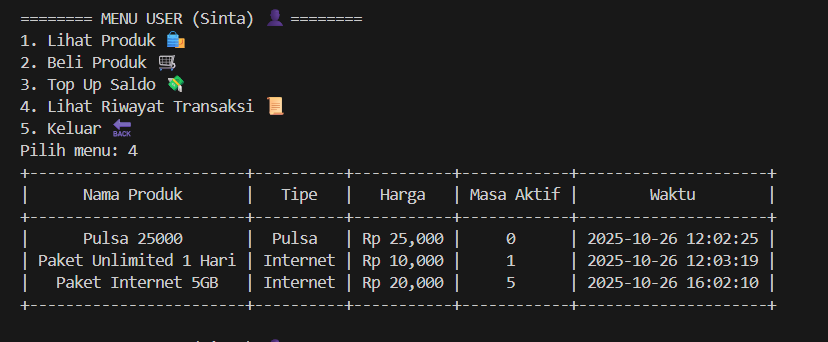
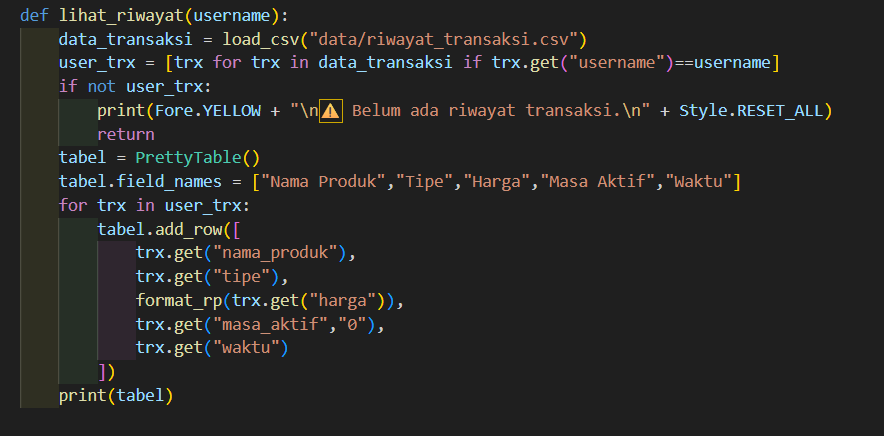
**Gambar 3.43** Alur Program: User – Top Up Saldo

A screenshot of a computer

AI-generated content may be incorrect.

**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

**Gambar 3.46** Hasil Terminal

**Gambar 3.45** Alur Program: User – Lihat Riwayat Transaksi

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

Penjelasan:  
Opsi Logout digunakan untuk keluar dari akun user dan kembali ke menu utama.A screenshot of a computer

AI-generated content may be incorrect. Hal ini menandakan berakhirnya sesi pengguna tanpa harus menutup program secara keseluruhan.

**Gambar 3.47** Alur Program: User – Logout

A screen shot of a computer

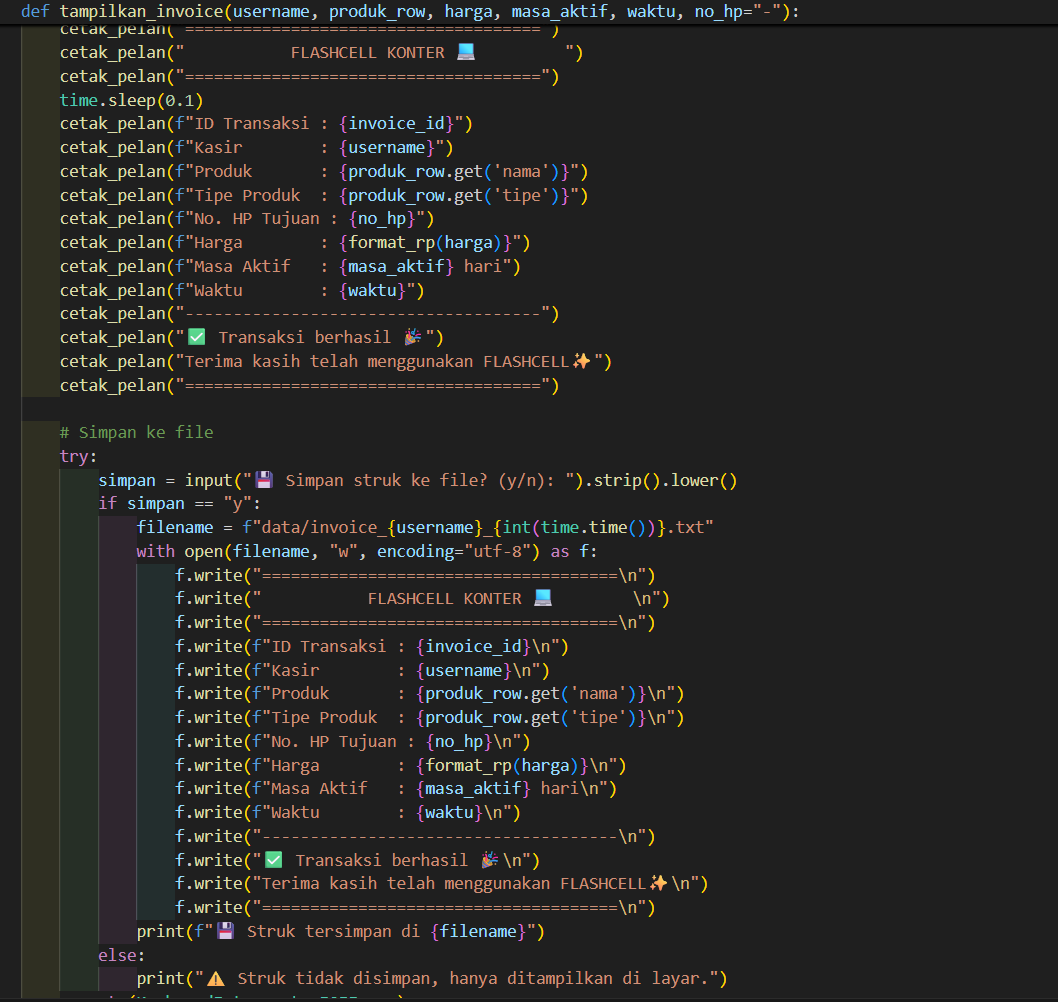
AI-generated content may be incorrect.A computer screen shot of a program code

AI-generated content may be incorrect.6. Pemilihan ID Produk

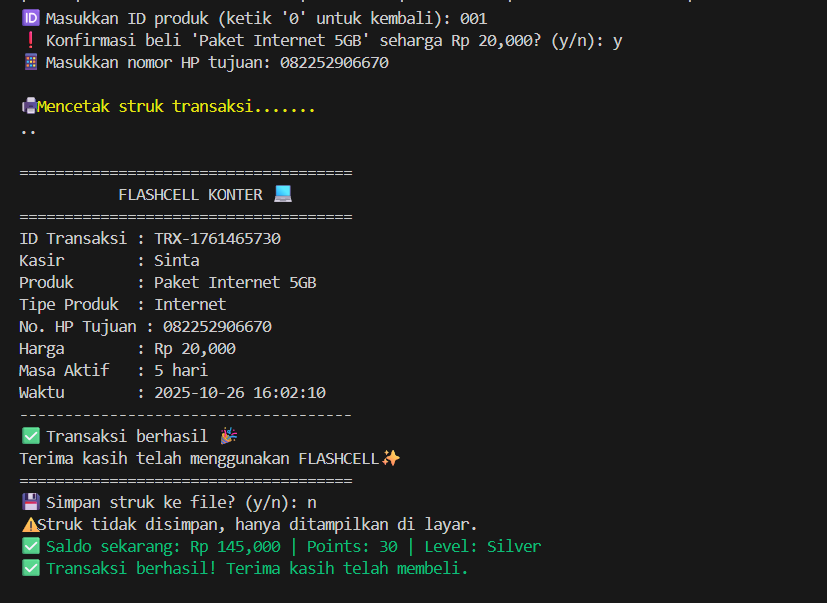
**Gambar 3.49** Hasil Terminal

**Gambar 3.48** Alur Program: Pemilihan ID Produk

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

**Gambar 3.50** Alur Program: Cetak Invoice Transaksi

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.

**Gambar 3.51** Hasil Terminal

8. Error Handling

A screen shot of a computer

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect.

**Gambar 3.53** Hasil Terminal

**Gambar 3.52** Alur Program: Error Handling

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.

A black background with white text

AI-generated content may be incorrect.

**Gambar 3.54** Penerapan: ValueError

2. File Handling Error

A screen shot of a computer code

AI-generated content may be incorrect.Fungsi load\_csv() menggunakan *try–except* untuk memastikan bahwa jika file belum ada, program tidak akan berhenti secara mendadak:

**Gambar 3.55** Penerapan: File Handling Error

3. KeyboardInterrupt & EOFError

A black background with white text

AI-generated content may be incorrect.Apabila pengguna menekan kombinasi *Ctrl + C*, program akan berhenti dengan pesan yang sopan tanpa menyebabkan error di terminal:

**Gambar 3.56** Penerapan: KeyboardInterrupt

A screen shot of a computer

AI-generated content may be incorrect.

**Gambar 3. 57** Hasil Terminal

A black screen with white text

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.**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).

A black screen with purple text

AI-generated content may be incorrect.A screen shot of a computer code

AI-generated content may be incorrect.9. Reminder Harian

**Gambar 3.61** Hasil Terminal

**Gambar 3.60** Alur Program: Reminder Harian

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.

A screenshot of a computer

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.10. Statistik Penjualan Admin

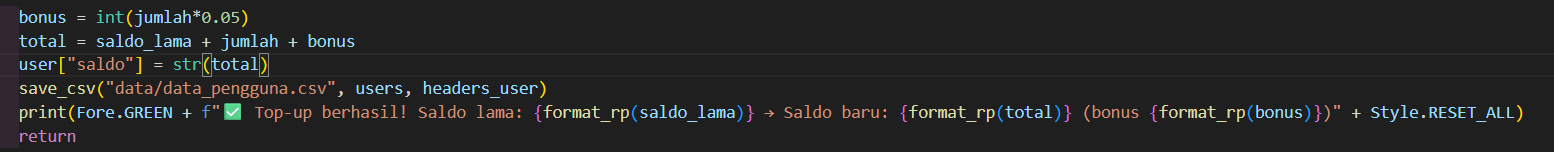
**Gambar 3.63** Hasil Terminal

**Gambar 3.62** Alur Program : Stastistik Admin

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.

A black screen with white text and green letters

AI-generated content may be incorrect.11. Bonus Top Up

**Gambar 3.65** Hasil Terminal

**Gambar 3.64** Alur Program : Bonus Top Up

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

|  |  |  |
| --- | --- | --- |
| **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 |

**Lampiran 1 : Tabel Kontribusi**

# A group of people pointing at a computer AI-generated content may be incorrect.

**Gambar 1.1** Lampiran: Konsul Ke 1

**Gambar 1.2** Lampiran: Konsul Ke 2