

APLIKASI MANAJEMEN TOKO GAME “RAYYSTORE GAMES” LAPORAN TUGAS STRUKTUR DATA

OLEH:

RAYYA SEVARENO

24416255201161



IF24D

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BUANA PERJUANGAN

KARAWANG 2025

DAFTAR ISI

DAFTAR ISI.....	2
DAFTAR GAMBAR	3
DAFTAR TABEL.....	5
BAB I PENDAHULUAN.....	6
1.1 Latar Belakang	6
1.2 Rumusan Masalah	7
1.3 Tujuan Proyek	7
BAB II METODOLOGI PERANCANGAN SOLUSI.....	8
2.1 Deskripsi Aplikasi	8
2.1.1 Gambaran Umum Alur Aplikasi	8
2.1.2. Fitur Utama yang Tersedia	9
2.1.3. Penjelasan Cara Kerja Aplikasi Secara Keseluruhan	13
2.1.4. Struktur File CSV.....	14
BAB III PERANCANGAN SOLUSI	16
3.1 Penjelasan Teknis	16
3.2 Hasil Uji Coba.....	26
a. Tampilan Menu	26
b. Contoh Input–Output.....	26
c. Pengujian Fungsional.....	32
d. Pengujian Integrasi.....	33
e. Catatan Kendala	33
BAB IV KESIMPULAN.....	34
DAFTAR PUSTAKA	36

DAFTAR GAMBAR

BAB II – Metodologi Perancangan Solusi

Gambar 2.1 Alur Keseluruhan Aplikasi	8
Gambar 2.2 Alur Menampilkan Produk Game.....	9
Gambar 2.3 Alur Menambahkan Produk Game.....	9
Gambar 2.4 Alur Memperbarui Produk Game.....	10
Gambar 2.5 Alur Menghapus Produk Game	10
Gambar 2.6 Alur Mencari Produk Game.....	11
Gambar 2.7 Alur Proses Transaksi Penjualan dan Pembelian.....	12
Gambar 2.8 Alur Proses Menampilkan Laporan Transaksi	13

BAB III – Perancangan Solusi

Modul fungsi_util.py

Gambar 3.1 Function load_produk	17
Gambar 3.2 Function simpan_produk.....	17
Gambar 3.3 Function load_transaksi.....	17
Gambar 3.4 Function simpan_transaksi.....	17

Modul fungsi_produk.py

Gambar 3.5 Function tampilkan_produk.....	18
Gambar 3.6 Function tambah_produk (1)	18
Gambar 3.7 Function tambah_produk (2)	18
Gambar 3.8 Function update_produk (1)	19
Gambar 3.9 Function update_produk (2)	19
Gambar 3.10 Function hapus_produk	19
Gambar 3.11 Function cari_produk.....	20

Modul fungsi_transaksi.py

Gambar 3.12 Function transaksi (1)	20
Gambar 3.13 Function transaksi (2).....	20
Gambar 3.14 Function transaksi (3).....	21
Gambar 3.15 Function transaksi (4).....	21

Modul laporan.py

Gambar 3.16 Function laporan_transaksi	21
--	----

Modul welcome.py

Gambar 3.17 Function welcome_message	22
--	----

Gambar 3.18 Function main	22
--	-----------

Bagian 3.2 Hasil Uji Coba

Gambar 3.19 Tampilan Menu	26
Gambar 3.20 Input-output Tambah Produk	26
Gambar 3.21 Hasil Tambah Produk (CSV)	26
Gambar 3.22 Input-output Update Produk	26
Gambar 3.23 Hasil Update Produk (CSV)	26
Gambar 3.24 Input-output Hapus Produk	26
Gambar 3.25 Hasil Hapus Produk (CSV)	27
Gambar 3.26 Input-output Cari Produk	27
Gambar 3.27 Input-output Transaksi Penjualan	27
Gambar 3.28 Input-output Transaksi Pembelian	27
Gambar 3.29 Hasil Transaksi di transaksi.csv	27
Gambar 3.30 Input-output Laporan Transaksi	27
Gambar 3.31 Output Laporan Transaksi	27

DAFTAR TABEL

Tabel 2.1 struktur produk.csv	14
Tabel 2.1 struktur transaksi.csv	15
Tabel 3.2 pengujian fungsional proyek	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam perkembangan era digital saat ini, proses pengelolaan data secara manual mulai digantikan oleh program digital yang jauh lebih mudah dan efisien. Salah satu sector yang juga memerlukan program digital adalah bidang manajemen penjualan toko dan distribusi game. Banyaknya jenis game yang tersedia serta tingginya minat masyarakat terhadap pembelian game menuntut adanya sistem manajemen data yang efisien, terutama bagi toko game yang menjual berbagai produk secara fisik maupun digital. Oleh karena itu, dibutuhkan suatu aplikasi manajemen jual beli game yang mampu membantu proses pengelolaan stok, transaksi penjualan dan pembelian, serta pelaporan penjualan dengan cara yang sistematis dan mudah digunakan.

Untuk mengatasi permasalahan tersebut, dikembangkanlah sebuah aplikasi manajemen penjualan toko game berbasis Python dengan memanfaatkan struktur data algoritma seperti array/list, queue, dan data searching. Aplikasi ini juga menggunakan file CSV sebagai sistem database untuk membuat, membaca, mengubah, menghapus data status manajemen beserta laporan transaksi secara permanen. Aplikasi ini menyediakan fitur pengelolaan produk (stok, harga, dan deskripsi), pencatatan transaksi penjualan dan pembelian, serta laporan penjualan harian, mingguan, dan bulanan. Proyek ini juga menjadi penerapan nyata dari konsep-konsep yang telah dipelajari dalam mata kuliah Struktur Data, seperti pengolahan array, antrian (queue), dan algoritma pencarian data.

Aplikasi ini penting karena dapat membantu pemilik atau pengelola toko game, khususnya skala kecil hingga menengah, dalam mencatat data secara lebih rapi dan terstruktur. Target pengguna dari aplikasi ini adalah pengelola toko game dan pemilik usaha kecil yang belum memiliki sistem manajemen data berbasis digital. Hasil yang ingin dicapai dari penggunaan aplikasi ini adalah terciptanya sistem manajemen data yang praktis, cepat, dan mampu menampilkan laporan penjualan secara otomatis.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam proyek ini adalah:

1. Bagaimana merancang konsep aplikasi manajemen toko game dalam pemrograman Python
2. Bagaimana mengimplementasikan struktur data array/list, queue, searching dalam mengelola produk game dan sistem transaksi
3. Bagaimana mengaplikasikan sistem database secara permanen menggunakan file CSV

1.3 Tujuan Proyek

Tujuan dari pembuatan proyek ini adalah:

1. Membangun aplikasi manajemen data jual beli untuk toko game yang dapat mengelola stok, transaksi, dan laporan.
2. Mengimplementasikan konsep struktur data algoritma seperti array/list, queue, dan searching dalam kasus nyata
3. Menggunakan file CSV sebagai penyimpanan data yang efisien dan mudah diakses.

BAB II

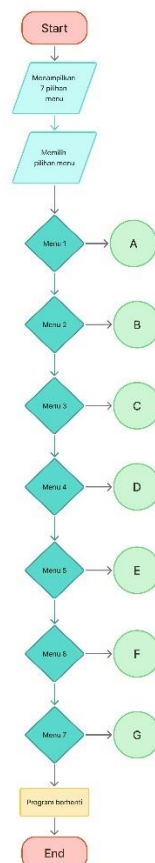
METODOLOGI PERANCANGAN SOLUSI

2.1 Deskripsi Aplikasi

Aplikasi yang dikembangkan bernama "RayyStore Games", yaitu sebuah sistem manajemen toko game berbasis terminal yang menggunakan file CSV (Comma-Separated Values) sebagai media penyimpanan data. Aplikasi ini dirancang untuk mengelola data produk, mencatat transaksi penjualan dan pembelian, serta menampilkan laporan penjualan berdasarkan periode waktu tertentu (harian, mingguan, dan bulanan).

2.1.1 Gambaran Umum Alur Aplikasi

Saat aplikasi dijalankan, pengguna akan disambut oleh tampilan menu utama berisi beberapa pilihan daftar fitur. Setiap menu akan menjalankan fungsi fitur sesuai kebutuhan pengguna. Alur ini dapat digambarkan dalam bentuk flowchart berikut:

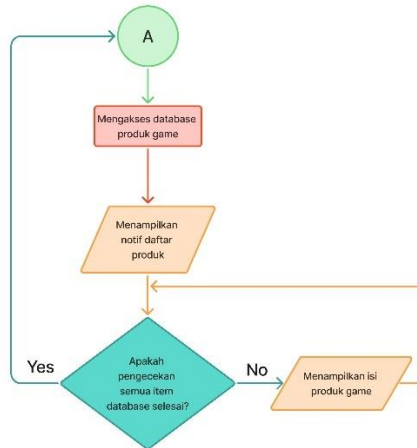


Gambar 2.1 Alur Keseluruhan Aplikasi

2.1.2. Fitur Utama yang Tersedia

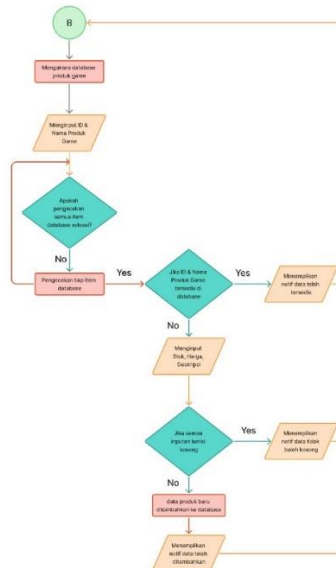
1. Manajemen Produk Game

- Menampilkan seluruh produk yang tersedia dalam file CSV.



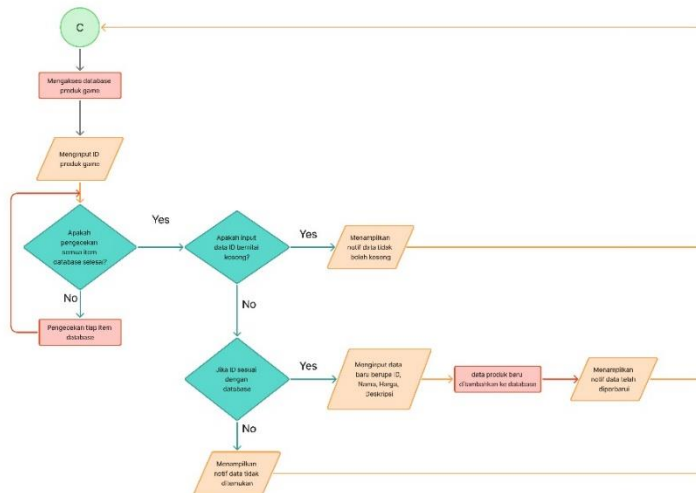
Gambar 2.1 Alur Menampilkan Produk Game

- Menambahkan produk baru dengan validasi ID dan nama agar tidak duplikat.



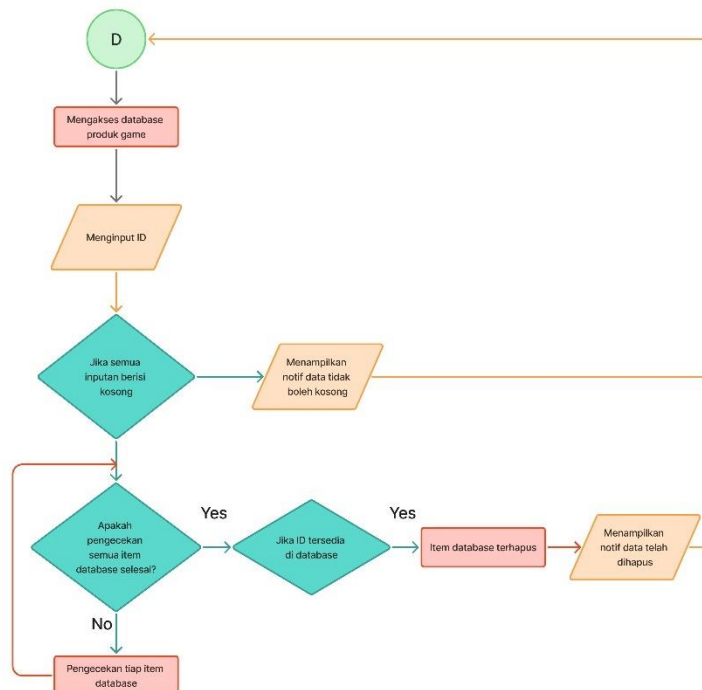
Gambar 2.1 Alur Menambahkan Produk Game

- Memperbarui data produk berdasarkan ID.



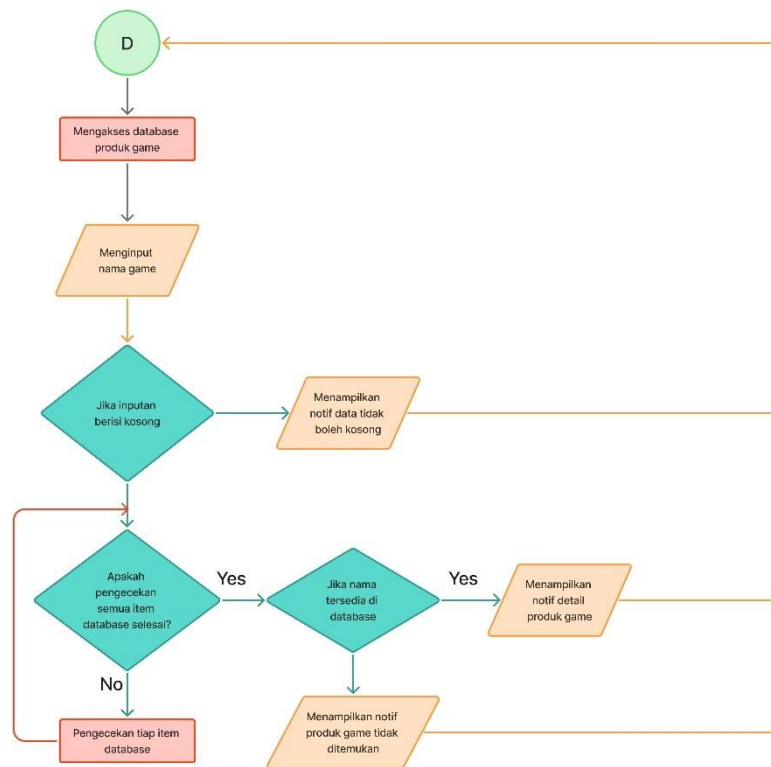
Gambar 2.1 Alur Memperbarui Produk Game

- Menghapus produk dari data.



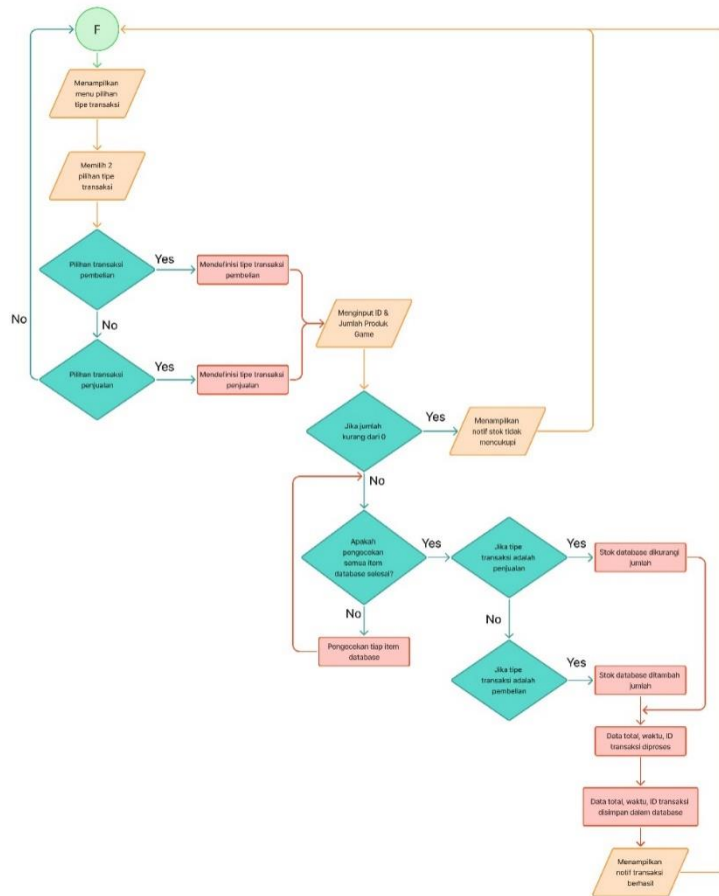
Gambar 2.1 Alur Menghapus Produk Game

- Mencari produk berdasarkan nama.



Gambar 2.1 Alur Mencari Produk Game

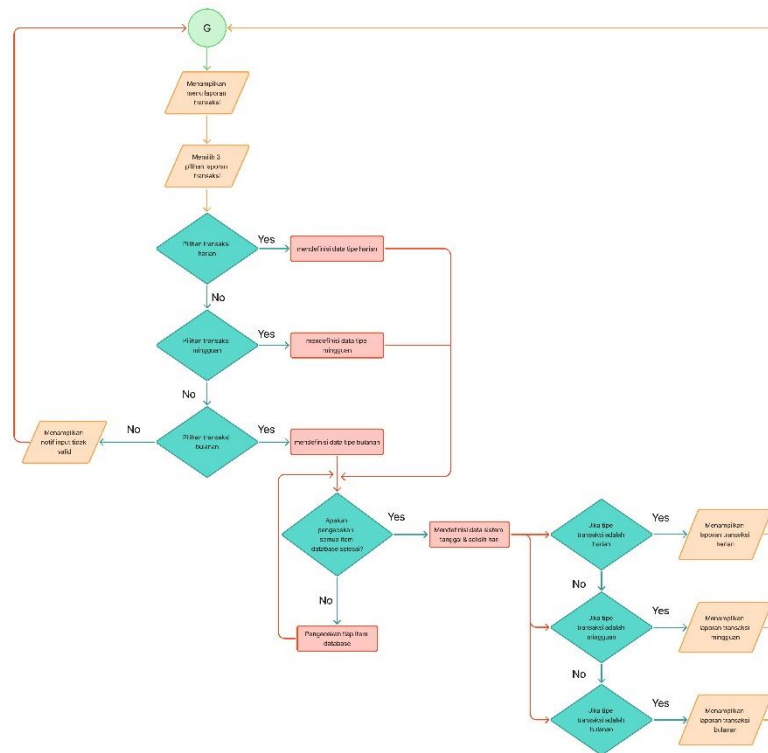
2. Transaksi



Gambar 2.1 Alur Proses Transaksi Penjualan dan Pembelian

- Melayani transaksi **penjualan** (mengurangi stok) dan **pembelian** (menambah stok).
- Validasi stok tersedia sebelum penjualan dilakukan.
- Perhitungan total harga berdasarkan jumlah dan harga produk.
- Mencatat transaksi ke file CSV dan queue.

3. Laporan Transaksi



Gambar 2.1 Alur Proses Menampilkan Laporan Transaksi

- Menyediakan laporan berdasarkan rentang waktu:
 - **Harian** (≤ 1 hari)
 - **Mingguan** (≤ 7 hari)
 - **Bulanan** (≤ 30 hari)
- Laporan ditampilkan dalam format yang memuat ID transaksi, tanggal, jumlah, total, dan tipe transaksi.

2.1.3. Penjelasan Cara Kerja Aplikasi Secara Keseluruhan

Secara keseluruhan, aplikasi ini bekerja melalui struktur beberapa fungsi yang kemudian dipisah menjadi beberapa file dan semua modular saling keterkaitan dengan file utama:

- **main.py** adalah pusat eksekusi program. Di dalamnya terdapat perulangan menu yang terus berjalan sampai pengguna memilih keluar. File ini menghubungkan semua fitur ke pengguna.

- **welcome.py** hanya menyimpan fungsi sambutan awal dengan daftar menu agar antarmuka lebih terstruktur.
- **fungsi_produk.py** berisi semua fungsi untuk pengelolaan produk seperti menambah, menampilkan, mengubah, menghapus, dan mencari data game yang disimpan di produk.csv. Semua input divalidasi agar tidak kosong, dan ID serta nama game tidak boleh duplikat saat menambah data.
- **fungsi_transaksi.py** menangani proses transaksi penjualan dan pembelian. Transaksi akan memvalidasi ketersediaan stok, menghitung total, serta mencatat transaksi tersebut ke dalam file transaksi.csv dan juga ke dalam *queue* (antrian data sementara).
- **laporan.py** mengelola pembuatan laporan transaksi berdasarkan waktu. File ini memproses data dari transaksi.csv dan memfilter transaksi berdasarkan hari, minggu, atau bulan.
- **fungsi_util.py** berisi fungsi pembantu seperti membaca (*load_produk*, *load_transaksi*) dan menyimpan data (*simpan_produk*, *simpan_transaksi*) dari dan ke file CSV.

Seluruh data produk dan transaksi disimpan secara lokal dalam file CSV, yang kemudian dimuat ulang saat dibutuhkan. Perubahan terhadap data seperti penambahan, penghapusan, dan pembaruan akan langsung ditulis ulang ke file CSV agar data tetap konsisten.

2.1.4. Struktur File CSV

1. File **produk.csv**

File ini menyimpan informasi tentang produk game. Kolom-kolom yang terdapat di dalamnya adalah:

Kolom	Keterangan
id	ID unik produk, biasanya berformat huruf+angka (contoh: P001)
nama	Nama game yang dijual
stok	Jumlah stok game yang tersedia
harga	Harga satuan dari produk game
deskripsi	Penjelasan singkat mengenai game

Tabel 2.1 struktur produk.csv

2. File **transaksi.csv**

File ini mencatat seluruh riwayat transaksi, baik pembelian maupun penjualan. Kolom-kolom yang disimpan:

Kolom	Keterangan
Tanggal	Tanggal transaksi dilakukan (YYYY-MM-DD)
id_transaksi	ID transaksi (contoh: T0001)
id_produk	ID dari produk yang dibeli/dijual
Jumlah	Jumlah unit yang dibeli/dijual
Game	Keterangan nama game
total	Total harga dari transaksi (harga x jumlah)
tipe	Jenis transaksi Penjualan atau Pembelian

Tabel 2.1 struktur transaksi.csv

Dengan pendekatan modular dan penyimpanan berbasis CSV, aplikasi ini dapat dijalankan secara ringan di berbagai perangkat tanpa database tambahan. Struktur dan alur kerja yang terorganisir juga memudahkan pemeliharaan serta pengembangan fitur di masa depan.

BAB III

PERANCANGAN SOLUSI

3.1 Penjelasan Teknis

Aplikasi *RayyStore Games* dikembangkan menggunakan bahasa Python dengan pendekatan modular. Struktur data utama menggunakan **list of dictionary** yang disimpan dan dimuat melalui file CSV. Setiap modul memiliki fungsi-fungsi khusus yang mendukung fitur aplikasi, seperti manajemen produk, transaksi, laporan, dan tampilan antarmuka terminal.

a. Struktur Data

Aplikasi menggunakan dua struktur data utama:

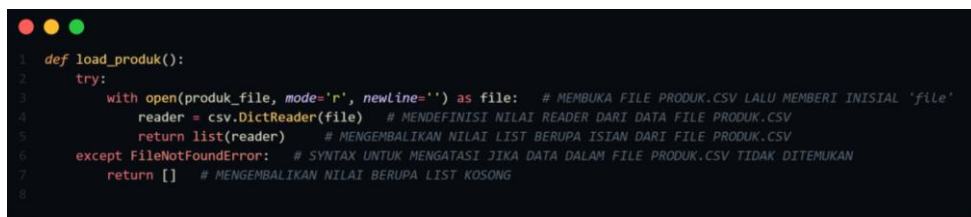
- **List of Dictionary** untuk menyimpan data dari file CSV.
- **Queue** (antrian) dari `collections.deque` untuk menyimpan transaksi sementara.
- **Searching** untuk mencari item produk game

b. Modul fungsi_util.py (Folder: fungsi_manajemen)

Modul ini bertanggung jawab memuat dan menyimpan data ke dalam file CSV.

- **load_produk()**

Membaca data dari produk.csv dan mengembalikan list of dict.



```
1 def load_produk():
2     try:
3         with open(produk_file, mode='r', newline='') as file: # MEMBUKA FILE PRODUK.CSV LALU MEMBERI INISIAL 'file'
4             reader = csv.DictReader(file) # MENDEFINISI NILAI READER DARI DATA FILE PRODUK.CSV
5             return list(reader) # MENGEMBALIKAN NILAI LIST BERUPA ISI DARI FILE PRODUK.CSV
6     except FileNotFoundError: # SYNTAX UNTUK MENGATASI JIKA DATA DALAM FILE PRODUK.CSV TIDAK DITEMUKAN
7         return [] # MENGEMBALIKAN NILAI BERUPA LIST KOSONG
8
```

Gambar 3.1 function load_produk

- **simpan_produk(data)**

Menulis ulang data produk ke produk.csv.

```

1 def simpan_produk(data):
2     with open(produk_file, mode='w', newline='') as file: # MEMBUKA FILE PRODUK.CSV LALU MEMBERI INISIAL 'file'
3         fieldnames = ['id', 'nama', 'stok', 'harga', 'deskripsi'] # MENDEFINISI KATEGORI YG DITAMBAHKAN PADA FILE CSV
4         writer = csv.DictWriter(file, fieldnames=fieldnames) # MEMBUAT OBJEK WRITER UNTUK MENULIS PADA FILE PRODUK.CSV
5         writer.writeheader() # MENULIS BARIS PERTAMA HEADER PADA PRODUK.CSV
6
7         # PERULANGAN UNTUK MELAKUKAN ITERASI SETIAP ITEM SEBAGAI BARIS BARU
8         for row in data:
9             writer.writerow(row) # MENULIS TERASI SEBAGAI SATU BARIS DATA
10

```

Gambar 3.1 function simpan_produk

- **load_transaksi()**

Membaca data dari transaksi.csv.

```

1 def load_transaksi():
2     try:
3         with open(transaksi_file, mode='r', newline='') as file: # MEMBUKA FILE TRANSAKSI.CSV LALU MEMBERI INISIAL 'file'
4             reader = csv.DictReader(file) # MENDEFINISI NILAI READER DARI DATA FILE TRANSAKSI.CSV
5             return list(reader) # MENGEMBALIKAN NILAI LIST BERUPA ISIAN DARI FILE TRANSAKSI.CSV
6     except FileNotFoundError: # SYNTAX UNTUK MENGATASI JIKA DATA DALAM FILE TRANSAKSI.CSV TIDAK DITEMUKAN
7         return [] # MENGEMBALIKAN NILAI BERUPA LIST KOSONG

```

Gambar 3.1 function load_transaksi

- **simpan_transaksi(data)**

Menulis ulang data transaksi ke transaksi.csv.

```

1 def simpan_transaksi(data):
2     with open(transaksi_file, mode='w', newline='') as file: # MEMBUKA FILE TRANSAKSI.CSV LALU MEMBERI INISIAL 'file'
3         fieldnames = ['tanggal', 'id_transaksi', 'id_produk', 'jumlah', 'game', 'total', 'tipe'] # MENDEFINISI KATEGORI YG DITAMBAHKAN PADA FILE CSV
4         writer = csv.DictWriter(file, fieldnames=fieldnames) # MEMBUAT OBJEK WRITER UNTUK MENULIS PADA FILE TRANSAKSI.CSV
5         writer.writeheader() # MENULIS BARIS PERTAMA HEADER PADA PRODUK.CSV
6
7         # PERULANGAN UNTUK MELAKUKAN ITERASI SETIAP ITEM SEBAGAI BARIS BARU
8         for row in data:
9             writer.writerow(row) # MENULIS TERASI SEBAGAI SATU BARIS DATA

```

Gambar 3.1 function simpan_transaksi

c. Modul fungsi_produk.py (Folder: fungsi_manajemen)

Modul ini menangani seluruh operasi CRUD terhadap data produk.

- **tampilkan_produk()**

Menampilkan daftar semua produk game yang tersimpan.



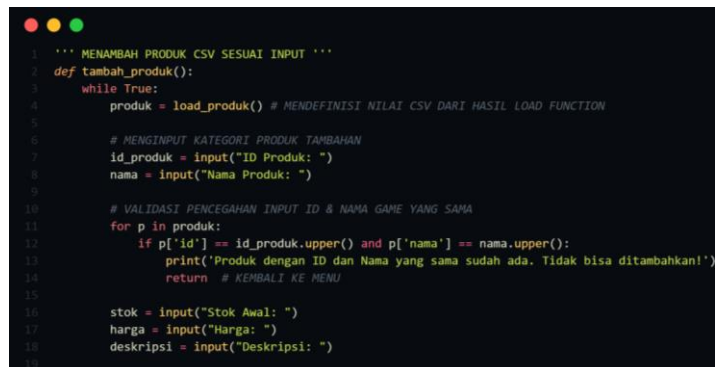
```
1 def tampilan_produk():
2     produk = load_produk() # MENDEFINISI NILAI CSV DARI HASIL LOAD FUNCTION
3     print("\n===== Daftar Produk Game =====")
4
5     # MENAMPILKAN ISI DATA CSV SECARA PERULANGAN BERUNTUNAN
6     for p in produk:
7         print(f"ID: {p['id']} | Nama: {p['nama']} | Stok: {p['stok']} | Harga: {p['harga']} | Deskripsi: {p['deskripsi']}")
8     print('=====')
```

Gambar 3.1 function tampilan_produk

- **tambah_produk()**

Menambahkan produk baru dengan validasi:

- ID dan nama tidak boleh duplikat
- Semua input wajib diisi (tidak boleh kosong)



```
1 ''' MENAMBAH PRODUK CSV SESUAI INPUT '''
2 def tambah_produk():
3     while True:
4         produk = load_produk() # MENDEFINISI NILAI CSV DARI HASIL LOAD FUNCTION
5
6         # MENGINPUT KATEGORI PRODUK TAMBAHAN
7         id_produk = input("ID Produk: ")
8         nama = input("Nama Produk: ")
9
10        # VALIDASI PENCEGAHAN INPUT ID & NAMA GAME YANG SAMA
11        for p in produk:
12            if p['id'] == id_produk.upper() and p['nama'] == nama.upper():
13                print('Produk dengan ID dan Nama yang sama sudah ada. Tidak bisa ditambahkan!')
14                return # KEMBALI KE MENU
15
16        stok = input("Stok Awal: ")
17        harga = input("Harga: ")
18        deskripsi = input("Deskripsi: ")
19
```

Gambar 3.1 function tambah_produk 1

```

20 # VALIDASI INPUT AGAR TIDAK TERJADI INPUT YANG KOSONG
21 if not id_produk.strip() or not nama.strip() or not stok.strip() or not harga.strip() or not deskripsi.strip():
22     print('Inputan tidak boleh kosong!')
23     return
24 else:
25     break # KELUAR PERULANGAN PROSES TAMBAH PRODUK GAME
26
27 # MENAMBAH HASIL INPUTAN KATEGORI SEMENTARA
28 produk.append({'id': id_produk.upper(), 'nama': nama.upper(), 'stok': stok, 'harga': harga, 'deskripsi': deskripsi.upper()})
29
30 simpan_produk(produk) # MENYIMPAN HASIL INPUTAN KATEGORI CSV LEWAT FUNCTION
31 print("Produk berhasil ditambahkan!")
32

```

Gambar 3.1 function tambah_produk 2

- **update_produk()**

Memperbarui data produk berdasarkan ID. Namun, implementasinya belum menghapus data lama, hanya menambahkan data baru.

```

1 ''' UPDATE PRODUK CSV SESUAI INPUT '''
2 def update_produk():
3     produk = load_produk() # MENDEFINISI NILAI CSV DARI HASIL LOAD FUNCTION
4
5     id_cari = input("Masukkan ID Produk: ")
6
7     # PERULANGAN BERUNTUN UNTUK VALIDASI URUTAN DATA PADA FILE CSV
8     for p in produk:
9
10        # VALIDASI INPUT AGAR TIDAK TERJADI INPUT YANG KOSONG
11        if not id_cari.strip():
12            print('Inputan tidak boleh kosong!')
13            return # KEMBALI KE MENU
14

```

Gambar 3.1 function update_produk 1

```

15 # VALIDASI INPUTAN DENGAN DATA YANG SESUAI
16 if p['id'] == id_cari.upper():
17     id_baru = input('Masukkan ID baru: ')
18     nama_baru = input('Masukkan Game baru: ')
19     harga_baru = input('Masukkan Harga baru: ')
20     deskripsi_baru = input('Masukkan Deskripsi baru: ')
21
22     # MEMPERBARUI ITEM DATABASE DENGAN INPUTAN
23     p['id'] = id_baru.upper()
24     p['nama'] = nama_baru.upper()
25     p['harga'] = harga_baru.upper()
26     p['deskripsi'] = deskripsi_baru.upper()
27
28     simpan_produk(produk) # MENYIMPAN HASIL INPUTAN KATEGORI CSV LEWAT FUNCTION
29     print('Produk Game berhasil diperbarui!')
30     return # KEMBALI KE MENU PROGRAM
31 print("Produk Game tidak ditemukan.")
32

```

Gambar 3.1 function update_produk 2

- **hapus_produk()**

Menghapus produk dari produk.csv berdasarkan ID.

```

1  ''' MENGHAPUS PRODUK CSV SESUAI INPUT '''
2  def hapus_produk():
3      produk = load_produk() # MENDEFINISI NILAI CSV DARI HASIL LOAD FUNCTION
4      id_hapus = input("ID Produk yang akan dihapus: ")
5
6      # VALIDASI INPUT AGAR TIDAK TERJADI INPUT YANG KOSONG
7      if not id_hapus.strip():
8          print('Inputan tidak boleh kosong!')
9          return # KEMBALI KE MENU
10
11     produk_baru = [p for p in produk if p['id'] != id_hapus.upper()] # PERULANGAN AGAR TIDAK TERJADI KESALAHAN HAPUS PRODUK GAME
12     simpan_produk(produk_baru) # MENYIMPAN HASIL INPUTAN KATEGORI CSV LEWAT FUNCTION
13
14     print("Produk Game dihapus!")
15

```

Gambar 3.1 function hapus_produk

- **cari_produk()**

Mencari dan menampilkan data game berdasarkan nama.

```

1  ''' MENCARI PRODUK CSV SESUAI INPUT '''
2  def cari_produk():
3      produk = load_produk() # MENDEFINISI NILAI CSV DARI HASIL LOAD FUNCTION
4
5      cari_produk = input('Masukkan game yang ingin dicari: ')
6
7      # PERULANGAN BERUNTUN UNTUK VALIDASI URUTAN DATA PADA FILE CSV
8      for p in produk:
9
10         # VALIDASI INPUT AGAR TIDAK TERJADI INPUT YANG KOSONG
11         if not cari_produk.strip():
12             print('Inputan tidak boleh kosong!')
13             return # KEMBALI KE MENU
14
15         # VALIDASI INPUTAN DENGAN DATA YANG DICARI
16         if cari_produk.upper() == p['id']:
17             print('Produk Game ditemukan!')
18             print('\nDetail Produk Game')
19             print(f'Nama Game: {p['nama']}\nID Game: {p['id']}\nStok: {p['stok']}\nDeskripsi: {p['deskripsi']}')
20             return
21     print('Produk Game tidak ditemukan!')

```

Gambar 3.1 function cari_produk

d. Modul fungsi_transaksi.py (Folder: fungsi_manajemen)

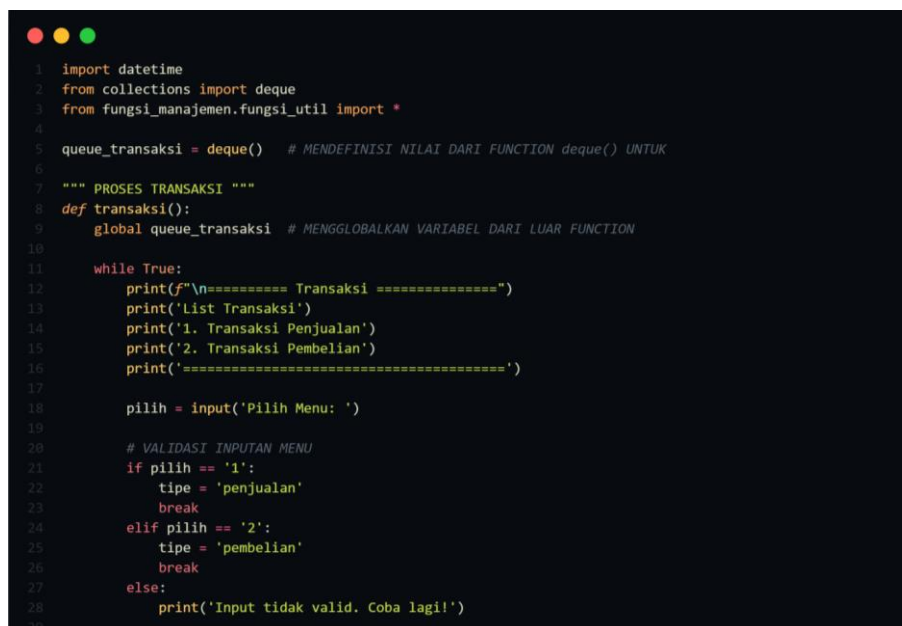
Modul ini menangani proses transaksi penjualan dan pembelian produk.

- **transaksi()**

Meminta pengguna memilih jenis transaksi:

- Jika penjualan, stok akan dikurangi
- Jika pembelian, stok akan ditambah

Transaksi disimpan ke transaksi.csv dan queue (deque) untuk antrian data sementara.

A screenshot of a code editor showing a Python script. The script is titled 'PROSES TRANSAKSI' and defines a function 'transaksi()'. It imports 'datetime' and 'deque' from 'collections', and imports '*' from 'fungsi_manajemen.fungsi_util'. It initializes 'queue_transaksi' as a deque. The function 'transaksi()' uses a global 'queue_transaksi' and enters a 'while True' loop. Inside the loop, it prints a menu with two options: '1. Transaksi Penjualan' and '2. Transaksi Pembelian'. It prompts the user to 'Pilih Menu: ' and then validates the input. If the user enters '1', it sets 'tipe' to 'penjualan'. If the user enters '2', it sets 'tipe' to 'pembelian'. If the input is invalid, it prints 'Input tidak valid. Coba lagi!'.

```
1 import datetime
2 from collections import deque
3 from fungsi_manajemen.fungsi_util import *
4
5 queue_transaksi = deque() # MENDEFINISI NILAI DARI FUNCTION deque() UNTUK
6
7 """ PROSES TRANSAKSI """
8 def transaksi():
9     global queue_transaksi # MENGGLOBAKAN VARIABEL DARI LUAR FUNCTION
10
11     while True:
12         print(f"\n===== Transaksi =====")
13         print('List Transaksi')
14         print('1. Transaksi Penjualan')
15         print('2. Transaksi Pembelian')
16         print('=====')
17
18         pilih = input('Pilih Menu: ')
19
20         # VALIDASI INPUTAN MENU
21         if pilih == '1':
22             tipe = 'penjualan'
23             break
24         elif pilih == '2':
25             tipe = 'pembelian'
26             break
27         else:
28             print('Input tidak valid. Coba lagi!')
```

Gambar 3.1 function transaksi 1

```

30 # SYNTAX INI BERFUNGSI MENGHINDARI KESALAHAN INPUT
31 try:
32
33     # PERULANGAN PROGRAM TRANSAKSI
34     while True:
35         produk = load_produk() # MENDEFINISI NILAI CSV DARI HASIL LOAD FUNCTION
36         transaksi = load_transaksi() # MENDEFINISI NILAI CSV DARI HASIL LOAD FUNCTION
37         id_produk = input("Masukkan ID Produk: ")
38         jumlah = int(input("Jumlah: "))
39
40         # VALIDASI JIKA INPUT JUMLAH KURANG DARI 0
41         if jumlah <= 0:
42             print("Jumlah tidak valid!")
43             return
44

```

Gambar 3.1 function transaksi 2

```

45 # PERULANGAN BERUNTUN UNTUK VALIDASI TRANSAKSI
46 for p in produk:
47
48     # VALIDASI UNTUK MENAMBAHKAN NAMA GAME KE LAPORAN TRANSAKSI
49     if p['id'] != p['nama']:
50         nama_game = p['nama']
51
52     # VALIDASI KESESUAIAN INPUT DENGAN KATEGORI ID CSV
53     if p['id'] == id_produk.upper():
54
55         # VALIDASI JIKA JUMLAH MELEBIHI STOK GAME
56         if tipe == 'penjualan' and int(p['stok']) < jumlah:
57             print("Stok tidak mencukupi!")
58             return
59

```

Gambar 3.1 function transaksi 3

```

60 # PROSES TRANSAKSI
61 if tipe == 'penjualan':
62     p['stok'] = str(int(p['stok']) - jumlah)
63 elif tipe == 'pembelian':
64     p['stok'] = str(int(p['stok']) + jumlah)
65
66 total = int(p['harga']) * jumlah
67 waktu = datetime.datetime.now().strftime('%Y-%m-%d')
68 id_trans = f"T{len(transaksi) + 1:04d}"
69
70 # DATA PENYIMPANAN SEMENTARA UNTUK DIKIRIM KE CSV
71 transaksi.append({'id_transaksi': id_trans.upper(), 'id_produk': id_produk.upper(),
72                 'tanggal': waktu, 'game': nama_game.upper(), 'jumlah': jumlah,
73                 'total': total, 'tipe': tipe.upper()})
74
75 # MENAMBAH DATA SEMENTARA KE ANTRIAN PERTAMA
76 queue_transaksi.append(transaksi[-1])
77
78 # MENYIMPAN / MEMINDAHKAN DATA SEMENTARA KE CSV LEWAT FUNCTION
79 simpan_produk(produk)
80 simpan_transaksi(transaksi)
81
82 print(f"Transaksi {tipe} berhasil!")
83 return
84 print("Produk tidak ditemukan.")
85 except ValueError: # SYNTAX INI BERFUNGSI MENGHINDARI KESALAHAN INPUT
86     print('Input tidak valid. Coba lagi!')

```

Gambar 3.1 function transaksi 4

e. Modul laporan.py

Modul ini menyajikan laporan transaksi berdasarkan periode:

- **laporan_transaksi()**

Memfilter transaksi berdasarkan waktu:

- Harian (selisih hari ≤ 1)
- Mingguan (≤ 7 hari)
- Bulanan (≤ 30 hari)

```
1 import datetime
2 from fungsi_manajemen.fungsi_util import load_transaksi
3
4 ''' LAPORAN TRANSAKSI '''
5 def laporan_transaksi():
6     transaksi = load_transaksi() # MENDEFINISI NILAI CSV DARI HASIL LOAD FUNCTION
7
8     # PERULANGAN PROGRAM LAPORAN
9     while True:
10         print(f"\n===== Laporan Transaksi =====")
11         hari_ini = datetime.datetime.now() # MENDEFINISI DATA WAKTU HARI INI
12
13         print('List Laporan Transaksi')
14         print('1. Transaksi Harian')
15         print('2. Transaksi Mingguan')
16         print('3. Transaksi Bulanan')
17         print('=====')
18
19         pilih = input('Pilih Transaksi: ')
20
21         # VALIDASI INPUTAN MENU
22         if pilih == '1':
23             tipe = 'harian'
24             break
25         elif pilih == '2':
26             tipe = 'mingguan'
27             break
28         elif pilih == '3':
29             tipe = 'bulanan'
30             break
31         else:
32             print('Input tidak valid. Coba lagi!')
33     print(f'\nLAPORAN TRANSAKSI {tipe.upper()}')
```

Gambar 3.1 function laporan_transaksi

f. Modul welcome.py (Folder: libs)

- **welcome_message()**

Menampilkan menu utama kepada pengguna. Ini merupakan antarmuka awal dari sistem CLI (command-line interface).

A screenshot of a code editor with a dark background and light-colored text. The code is a Python function named `welcome_message()` that prints a menu for a game store. The menu is titled "PESAN MENU" and lists seven options: 1. Tampilkan Produk Game, 2. Tambah Produk Game, 3. Update Produk Game, 4. Hapus Produk Game, 5. Cari Produk Game, 6. Transaksi, and 7. Laporan Transaksi. The function also prints a separator line of equals signs. The code is as follows:

```
1  ''' PESAN MENU '''
2  def welcome_message():
3      print("\n===== RAYYSTORE GAMES =====")
4      print('List Menu:')
5      print("1. Tampilkan Produk Game")
6      print("2. Tambah Produk Game")
7      print("3. Update Produk Game")
8      print("4. Hapus Produk Game")
9      print("5. Cari Produk Game")
10     print("6. Transaksi")
11     print("7. Laporan Transaksi")
12     print("0. Keluar")
13     print('=====')
```

Gambar 3.1 function welcome_message

g. Modul main.py

Modul utama ini mengatur alur aplikasi. Fungsi menu() melakukan loop tak hingga untuk menerima input pengguna dan memanggil fungsi sesuai menu.

```
1  from libs.welcome import welcome_message
2  from fungsi_manajemen.fungsi_produk import *
3  from fungsi_manajemen.fungsi_transaksi import transaksi
4  from laporan import laporan_transaksi
5
6  ''' MAIN PROGRAM '''
7  def menu():
8      while True: # REKURSI PERULANGAN WHILE
9          welcome_message() # MENAMPILKAN OUTPUT SAMBUTAN
10         pilih = input("Pilih menu: ")
11
12         # VALIDASI INPUT PILIHAN MENU
13         if pilih == '1':
14             tampilkan_produk()
15         elif pilih == '2':
16             tambah_produk()
17         elif pilih == '3':
18             update_produk()
19         elif pilih == '4':
20             hapus_produk()
21         elif pilih == '5':
22             cari_produk()
23         elif pilih == '6':
24             transaksi()
25         elif pilih == '7':
26             laporan_transaksi()
27         elif pilih == '0':
28             print("Terima kasih!")
29             break # PROGRAM BERHENTI
30         else:
31             print("Pilihan tidak valid.") # MENU PROGRAM BERULANG DARI AWAL
32
33 if __name__ == '__main__':
34     menu()
```

Gambar 3.1 function main

3.2 Hasil Uji Coba

a. Tampilan Menu

```
===== RAYYSTORE GAMES =====  
List Menu:  
1. Tampilkan Produk Game  
2. Tambah Produk Game  
3. Update Produk Game  
4. Hapus Produk Game  
5. Cari Produk Game  
6. Transaksi  
7. Laporan Transaksi  
0. Keluar  
=====  
Pilih menu: █
```

Gambar 3.2 tampilan menu

b. Contoh Input–Output

1. Tambah Produk Game

```
===== RAYYSTORE GAMES =====  
List Menu:  
1. Tampilkan Produk Game  
2. Tambah Produk Game  
3. Update Produk Game  
4. Hapus Produk Game  
5. Cari Produk Game  
6. Transaksi  
7. Laporan Transaksi  
0. Keluar  
=====  
Pilih menu: 2  
ID Produk: a011  
Nama Produk: gitar hero 2  
Stok Awal: 10  
Harga: 22000  
Deskripsi: Music  
Produk berhasil ditambahkan!
```

Gambar 3.2 tampilan input-output tambah produk game

```

produk.csv
1  id,nama,stok,harga,deskripsi
2  R001,GOD OF WAR RAGNAROK,39,105000000,ACTION
3  R002,MORTAL KOMBAT 11,30,599000,ACTION
4  R003,MINECRAFT,51,109500,SURVIVAL
5  R004,CRY OF FEAR,20,20000,HORROR
6  R005,OMORI,151,194000,STORY
7  R006,UNDERTALE,11,256000,STORY
8  R007,CALL OF DUTY 2: MODERN WARFARE,10,60000,MILITARY
9  R008,GTA V,55,259000,STORY
10 R009,FIFA 25,31,488000,SPORT
11 R010,A SPACE FOR THE UNBOUND,30,389000,STORY
12 A011,GUITAR HERO 2,10,22000,MUSIC
13

```

Gambar 3.2 hasil tambahan produk di file produk.csv

2. Update Produk Game

```

===== RAYYSTORE GAMES =====
List Menu:
1. Tampilkan Produk Game
2. Tambah Produk Game
3. Update Produk Game
4. Hapus Produk Game
5. Cari Produk Game
6. Transaksi
7. Laporan Transaksi
0. Keluar

=====
Pilih menu: 3
Masukkan ID Produk: a011
Masukkan ID baru: a001
Masukkan Game baru: man of medan
Masukkan Harga baru: 599000
Masukkan Deskripsi baru: horror
Produk Game berhasil diperbarui!

```

Gambar 3.2 tampilan input-output update produk game

```
produk.csv
1 id,nama,stok,harga,deskripsi
2 R001,GOD OF WAR RAGNAROK,39,105000000,ACTION
3 R002,MORTAL KOMBAT 11,30,599000,ACTION
4 R003,MINECRAFT,51,109500,SURVIVAL
5 R004,CRY OF FEAR,20,20000,HORROR
6 R005,OMORI,151,194000,STORY
7 R006,UNDERTALE,11,256000,STORY
8 R007,CALL OF DUTY 2: MODERN WARFARE,10,60000,MILITARY
9 R008,GTA V,55,259000,STORY
10 R009,FIFA 25,31,488000,SPORT
11 R010,A SPACE FOR THE UNBOUND,30,389000,STORY
12 A001,MAN OF MEDAN,10,599000,HORROR
13
```

Gambar 3.2 hasil pengubahan produk di file produk.csv

3. Hapus Produk Game

```
===== RAYYSTORE GAMES =====
List Menu:
1. Tampilkan Produk Game
2. Tambah Produk Game
3. Update Produk Game
4. Hapus Produk Game
5. Cari Produk Game
6. Transaksi
7. Laporan Transaksi
0. Keluar

=====
Pilih menu: 4
ID Produk yang akan dihapus: a001
Produk Game dihapus!
```

Gambar 3.2 tampilan input-output hapus produk game

```

produk.csv
1  id,nama,stok,harga,deskripsi
2  R001,GOD OF WAR RAGNAROK,39,105000000,ACTION
3  R002,MORTAL KOMBAT 11,30,599000,ACTION
4  R003,MINECRAFT,51,109500,SURVIVAL
5  R004,CRY OF FEAR,20,20000,HORROR
6  R005,OMORI,151,194000,STORY
7  R006,UNDERTALE,11,256000,STORY
8  R007,CALL OF DUTY 2: MODERN WARFARE,10,60000,MILITARY
9  R008,GTA V,55,259000,STORY
10 R009,FIFA 25,31,488000,SPORT
11 R010,A SPACE FOR THE UNBOUND,30,389000,STORY
12

```

Gambar 3.2 hasil penghapusan produk di file produk.csv

4. Cari Produk Game

```

===== RAYYSTORE GAMES =====
List Menu:
1. Tampilkan Produk Game
2. Tambah Produk Game
3. Update Produk Game
4. Hapus Produk Game
5. Cari Produk Game
6. Transaksi
7. Laporan Transaksi
0. Keluar
=====
Pilih menu: 5
Masukkan game yang ingin dicari: a001
Produk Game ditemukan!

Detail Produk Game
Nama Game: MAN OF MEDAN
ID Game: A001
Stok: 10
Deskripsi: HORROR

```

Gambar 3.2 tampilan input-output cari produk game

5. Transaksi

```

1. Tampilkan Produk Game
2. Tambah Produk Game
3. Update Produk Game
4. Hapus Produk Game
5. Cari Produk Game
6. Transaksi
7. Laporan Transaksi
0. Keluar
=====
Pilih menu: 6

===== Transaksi =====
List Transaksi
1. Transaksi Penjualan
2. Transaksi Pembelian
=====
Pilih Menu: 1
Masukkan ID Produk: r001
Jumlah: 4
Transaksi penjualan berhasil!

```

Gambar 3.2 tampilan input-output transaksi penjualan

```

1. Tampilkan Produk Game
2. Tambah Produk Game
3. Update Produk Game
4. Hapus Produk Game
5. Cari Produk Game
6. Transaksi
7. Laporan Transaksi
0. Keluar
=====
Pilih menu: 6

===== Transaksi =====
List Transaksi
1. Transaksi Penjualan
2. Transaksi Pembelian
=====
Pilih Menu: 2
Masukkan ID Produk: r003
Jumlah: 7
Transaksi pembelian berhasil!

```

Gambar 3.2 tampilan input-output transaksi pembelian

```

18 2025-07-12,r001,r001,4,GOD OF WAR RAGNAROK,42000000,PENJUALAN
19 2025-07-12,T0018,R001,4,GOD OF WAR RAGNAROK,42000000,PENJUALAN
20 2025-07-12,T0019,R003,7,MINECRAFT,766500,PEMBELIAN
21

```

Gambar 3.2 tampilan hasil transaksi pada transaksi.csv

6. Laporan Transaksi

```
===== RAYYSTORE GAMES =====
List Menu:
1. Tampilkan Produk Game
2. Tambah Produk Game
3. Update Produk Game
4. Hapus Produk Game
5. Cari Produk Game
6. Transaksi
7. Laporan Transaksi
0. Keluar

=====
Pilih menu: 7

===== Laporan Transaksi =====
List Laporan Transaksi
1. Transaksi Harian
2. Transaksi Mingguan
3. Transaksi Bulanan

=====
Pilih Transaksi: 1
```

Gambar 3.2 tampilan input-output laporan transaksi

LAPORAN TRANSAKSI MINGGUAN									
Tanggal: 2025-07-07	ID Transaksi: T0001	ID Produk: R001	Jumlah Mingguan: 2	Game: GOD OF WAR RAGNAROK	Total: 6000	Tipe: PENJUALAN			
Tanggal: 2025-07-07	ID Transaksi: T0002	ID Produk: R002	Jumlah Mingguan: 1	Game: MORTAL KOMBAT 11	Total: 5000	Tipe: PEMBELIAN			
Tanggal: 2025-07-07	ID Transaksi: T0003	ID Produk: R002	Jumlah Mingguan: 10	Game: MORTAL KOMBAT 11	Total: 50000	Tipe: PENJUALAN			
Tanggal: 2025-07-07	ID Transaksi: T0004	ID Produk: R002	Jumlah Mingguan: 10	Game: MORTAL KOMBAT 11	Total: 50000	Tipe: PEMBELIAN			
Tanggal: 2025-07-07	ID Transaksi: T0005	ID Produk: R003	Jumlah Mingguan: 12	Game: MINECRAFT	Total: 60000	Tipe: PENJUALAN			
Tanggal: 2025-07-09	ID Transaksi: T0006	ID Produk: R004	Jumlah Mingguan: 11	Game: CRY OF FEAR	Total: 99000	Tipe: PEMBELIAN			
Tanggal: 2025-07-09	ID Transaksi: T0007	ID Produk: R004	Jumlah Mingguan: 11	Game: CRY OF FEAR	Total: 99000	Tipe: PENJUALAN			
Tanggal: 2025-07-09	ID Transaksi: T0008	ID Produk: R001	Jumlah Mingguan: 1	Game: GOD OF WAR RAGNAROK	Total: 9000	Tipe: PENJUALAN			
Tanggal: 2025-07-10	ID Transaksi: T0009	ID Produk: R006	Jumlah Mingguan: 1	Game: UNDERTALE	Total: 120000	Tipe: PENJUALAN			
Tanggal: 2025-07-10	ID Transaksi: T0010	ID Produk: R006	Jumlah Mingguan: 1	Game: UNDERTALE	Total: 120000	Tipe: PEMBELIAN			
Tanggal: 2025-07-10	ID Transaksi: T0011	ID Produk: R001	Jumlah Mingguan: 10	Game: GOD OF WAR RAGNAROK	Total: 8900000	Tipe: PEMBELIAN			
Tanggal: 2025-07-10	ID Transaksi: T0012	ID Produk: R006	Jumlah Mingguan: 1	Game: UNDERTALE	Total: 120000	Tipe: PENJUALAN			
Tanggal: 2025-07-11	ID Transaksi: T0013	ID Produk: R005	Jumlah Mingguan: 1	Game: OMORI	Total: 100000	Tipe: PEMBELIAN			
Tanggal: 2025-07-12	ID Transaksi: T0014	ID Produk: R001	Jumlah Mingguan: 1	Game: GOD OF WAR RAGNAROK	Total: 10500000	Tipe: PENJUALAN			
Tanggal: 2025-07-12	ID Transaksi: T0015	ID Produk: R003	Jumlah Mingguan: 1	Game: MINECRAFT	Total: 109500	Tipe: PEMBELIAN			
Tanggal: 2025-07-12	ID Transaksi: T0016	ID Produk: R010	Jumlah Mingguan: 5	Game: A SPACE FOR THE UNBOUND	Total: 1945000	Tipe: PEMBELIAN			

Gambar 3.2 output laporan transaksi

c. Pengujian Fungsional

Fitur	Status	Keterangan
Tambah Produk	Berhasil	Validasi input dan penulisan ke CSV berjalan baik
Update Produk	Berhasil	Produk diubah dari CSV sesuai inputan pengguna
Hapus Produk	Berhasil	Produk dihapus dari CSV sesuai ID
Cari Produk	Berhasil	Mencari berdasarkan nama game
Transaksi Penjualan	Berhasil	Stok dikurangi dan transaksi disimpan
Transaksi Pembelian	Berhasil	Stok ditambah dan transaksi dicatat
Laporan Harian/Mingguan/Bulanan	Berhasil	Filter berdasarkan tanggal berfungsi

Tabel 3.2 pengujian fungsional proyek

d. Pengujian Integrasi

- Semua modul saling terhubung dengan baik melalui main.py.
- Uji coba skenario lengkap:
 1. Tambah produk
 2. Lakukan transaksi
 3. Lihat laporan→ semua berjalan tanpa error dan file CSV terupdate otomatis.

e. Catatan Kendala

- **Validasi jumlah:** belum ada pengecekan apakah input jumlah bertipe integer sebelum diproses.
- **Rollback:** belum ada penanganan jika proses transaksi gagal di tengah-tengah (misalnya saat file gagal dibuka).

BAB IV

KESIMPULAN

Aplikasi RayyStore Games telah berhasil dikembangkan sebagai sistem manajemen toko game berbasis *command-line interface* (CLI) dengan memanfaatkan file CSV sebagai media penyimpanan data. Aplikasi ini mampu menjalankan berbagai fungsi penting seperti pengelolaan produk (tambah, ubah, hapus, cari), pencatatan transaksi (penjualan dan pembelian), serta penyajian laporan transaksi berdasarkan periode waktu harian, mingguan, dan bulanan.

Berdasarkan hasil uji coba dan analisis fungsionalitas, aplikasi menunjukkan kinerja yang stabil dan dapat dijalankan tanpa error selama semua input diberikan sesuai format. Namun, terdapat beberapa hal yang dapat dijadikan rekomendasi pengembangan untuk versi mendatang, antara lain:

1. **Peningkatan Fitur Update Produk**

Perlu dilakukan revisi logika update agar benar-benar mengganti data lama, bukan hanya menambahkan data baru. Mekanisme pencarian dan penghapusan item lama berdasarkan ID sebaiknya ditambahkan.

2. **Validasi Tipe Data Input**

Pengguna dapat memasukkan data non-numerik pada kolom seperti stok, harga, dan jumlah transaksi. Disarankan untuk menambahkan pengecekan tipe data (misal `isdigit()` atau `try-except`) agar lebih aman dan akurat.

3. **Rollback Otomatis saat Transaksi Gagal**

Saat transaksi gagal di tengah jalan, sebaiknya sistem bisa membatalkan proses secara otomatis dan tidak menyimpan perubahan parsial ke file CSV.

4. **Antarmuka yang Lebih Ramah Pengguna**

Pengembangan antarmuka grafis (GUI) sederhana menggunakan library seperti `tkinter` atau `PyQt` dapat dipertimbangkan untuk meningkatkan pengalaman pengguna.

5. Penerapan Fitur Pencarian dan Filter Lanjutan

Filter berdasarkan rentang tanggal, nama game, atau jenis transaksi bisa ditambahkan untuk meningkatkan fleksibilitas laporan.

6. Penggunaan Database

Untuk skala yang lebih besar dan sistematis, disarankan mengganti penyimpanan berbasis file CSV dengan sistem basis data seperti SQLite atau PostgreSQL, agar aplikasi lebih efisien dan aman dalam menyimpan data.

7. Penambahan Fitur Login/Admin

Agar lebih aman, aplikasi dapat dilengkapi sistem autentikasi sederhana untuk membedakan peran pengguna (admin vs staf biasa), terutama saat menghapus atau mengubah data.

Dengan menerapkan rekomendasi tersebut, diharapkan aplikasi RayyStore Games dapat berkembang menjadi sistem manajemen penjualan yang lebih lengkap, fleksibel, dan andal, serta mampu mendukung kebutuhan operasional toko game digital dalam skala yang lebih besar di masa depan.

DAFTAR PUSTAKA

[1] Python Software Foundation, “csv — CSV File Reading and Writing,” *Python 3.12 Documentation*. [Online]. Available: <https://docs.python.org/3/library/csv.html>.

[2] Python Software Foundation, “collections — Container datatypes,” *Python 3.12 Documentation*. [Online]. Available: <https://docs.python.org/3/library/collections.html#collections.deque>.