

MAKALAH
DASAR DASAR PEMROGRAMAN
SISTEM PENJUALAN MERCHANDISE
KOMUNITAS



Disusun Oleh:
KELOMPOK C4

Farrel Fazzadinata Sumarna	2509116111
Dliya Syahla Hariyanto	2509116095
Riko Azkarazendra	2509116104
Russiananto	

Asisten Laboratorium:

Dwi Pebriyanto Pradana
2409116012

Taufik Ramadhani
2409116001

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
2025

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya, sehingga kami dapat menyelesaikan laporan proyek akhir praktikum Dasar-Dasar Pemrograman ini dengan baik. Laporan dengan judul “Sistem Penjualan Merchandise Komunitas” ini kami susun sebagai bentuk tugas akhir dari proyek yang telah kami kerjakan. Meskipun pada awalnya kami mengalami beberapa kesulitan dalam memahami konsep logika pemrograman dan pembagian tugas, namun dengan kerja sama tim yang baik serta bimbingan dari asisten laboratorium, akhirnya kami dapat menyelesaikan proyek ini dengan lancar.

Dalam laporan ini, kami berusaha menjelaskan secara rinci proses pembuatan sistem, mulai dari penulisan kode program, rancangan alur kerja sistem melalui flowchart, hingga penjelasan fungsi-fungsi yang terdapat di dalam program. Kami berharap laporan ini dapat memberikan pemahaman yang jelas mengenai cara kerja sistem penjualan merchandise komunitas berbasis Python yang telah kami buat.

Kami menyadari bahwa laporan ini masih jauh dari sempurna dan mungkin masih memiliki kekurangan. Oleh karena itu, kami sangat terbuka terhadap kritik dan saran yang membangun demi perbaikan di masa mendatang.

Akhir kata, kami mengucapkan terima kasih kepada dosen pengampu, asisten laboratorium, dan teman-teman yang telah memberikan dukungan serta bimbingan selama proses pengerjaan proyek ini.

Samarinda, 26 Oktober, 2025

Penyusun

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iii
DAFTAR TABEL	iv
DAFTAR LAMPIRAN.....	v
BAB I PENDAHULUAN	1
1.1 Deskripsi Masalah.....	1
1.2 Rumusan Masalah.....	1
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Manfaat	2
BAB II PERANCANGAN.....	3
2.1 Analisis Program	3
2.2 Flowchart.....	5
BAB III HASIL DAN PEMBAHASAN.....	8
3.1 Implementasi Program.....	8
3.2 Alur Program	10
3.3 Source Code.....	19
BAB IV PENUTUP.....	32
4.1 Kesimpulan.....	32
4.2 Saran	33
DAFTAR PUSTAKA	34
LAMPIRAN	35

DAFTAR GAMBAR

Gambar 2. 1 Flowchart Menu Utama	5
Gambar 2. 2 Flowchart Menu admin	6
Gambar 2. 3 Flowchart Menu User.....	7
Gambar 3. 1 Alur program Menu Utama.....	11
Gambar 3. 2 buat akun user	12
Gambar 3. 3 login user.....	12
Gambar 3. 4 Menu User.....	13
Gambar 3. 5 beli produk	13
Gambar 3. 6 Top up saldo.....	14
Gambar 3. 7 logout	14
Gambar 3. 8 menu admin.....	15
Gambar 3. 9 lihat produk	15
Gambar 3. 10 Tambah Produk	16
Gambar 3. 11 Tambah Produk	17
Gambar 3. 12 hapus produk	18
Gambar 3. 13 logout	18

DAFTAR TABEL

Tabel 3. 1 Source Code	19
Tabel 3. 2 Source Code	20
Tabel 3. 3 Source Code	21
Tabel 3. 4 Source Code	22
Tabel 3. 5 Source Code	23
Tabel 3. 6 Source Code	24
Tabel 3. 7 Source Code	25
Tabel 3. 8 Source Code	26
Tabel 3. 9 Source Code	27
Tabel 3. 10 Source Code	28
Tabel 3. 11 Source Code	29
Tabel 3. 12 Source Code	30
Tabel 3. 13 Source Code	31

DAFTAR LAMPIRAN

Lampiran 1 : Tabel Kontribusi.....	35
------------------------------------	----

BAB I

PENDAHULUAN

1.1 Deskripsi Masalah

Di era digital saat ini, kegiatan jual beli sudah banyak dilakukan secara online. Namun, masih banyak komunitas kecil atau organisasi yang menjual merchandise mereka secara manual, seperti mencatat transaksi di buku atau di Excel. Hal ini sering menyebabkan kesalahan pencatatan, kehilangan data, serta sulitnya mengelola stok barang dan menghitung saldo dengan akurat. Oleh karena itu, kami membuat sistem penjualan merchandise komunitas berbasis Python yang dapat membantu admin mengelola data produk, melakukan transaksi pembelian, serta memberikan kemudahan bagi pengguna untuk membeli produk dan mengatur saldo mereka.

Melalui mata kuliah Dasar-Dasar Pemrograman, kami mempelajari konsep input-output, perulangan, percabangan, serta penggunaan file eksternal. Dengan konsep tersebut, kami mencoba mengaplikasikan pemahaman kami dalam bentuk proyek nyata berupa sistem penjualan merchandise komunitas. Program ini diharapkan dapat menjadi solusi sederhana untuk mempermudah proses transaksi, pengelolaan stok, serta pembukuan saldo pengguna.

1.2 Rumusan Masalah

Berdasarkan deskripsi masalah di atas, rumusan masalah yang ingin diselesaikan adalah:

1. Bagaimana cara membuat sistem penjualan merchandise yang mudah digunakan oleh anggota komunitas?
2. Bagaimana sistem dapat mencatat data produk dan transaksi dengan akurat?
3. Bagaimana sistem dapat membantu membuat laporan penjualan dan stok produk secara cepat?

1.3 Batasan Masalah

Agar proyek ini fokus dan dapat diselesaikan, batasan masalah yang ditetapkan adalah:

1. Sistem hanya digunakan untuk penjualan merchandise dalam komunitas tertentu.
2. Sistem mencakup fitur: pencatatan produk, transaksi penjualan, dan laporan stok.
3. Sistem dibuat berbasis Python dengan tampilan sederhana di terminal/*console*.
4. Sistem tidak terhubung dengan pembayaran online atau sistem inventori kompleks lainnya.

1.4 Tujuan

Adapun tujuan dari pembuatan sistem ini adalah:

1. Membantu anggota komunitas dalam mencatat produk dan transaksi penjualan.
2. Mempermudah pengelolaan stok merchandise agar lebih terstruktur.
3. Menyediakan laporan penjualan secara cepat dan akurat.

1.5 Manfaat

Manfaat dari sistem ini antara lain:

1. Meminimalisir kesalahan pencatatan manual.
2. Menghemat waktu dalam proses transaksi dan pembuatan laporan.
3. Memberikan kemudahan bagi anggota komunitas dalam mengelola penjualan merchandise.

BAB II

PERANCANGAN

2.1 Analisis Program

Program ini dibuat menggunakan bahasa pemrograman Python dengan beberapa library tambahan untuk mempermudah pengelolaan data, tampilan tabel, dan keamanan input. Berikut analisis komponen yang digunakan:

1. Library CSV

- File user.csv digunakan untuk menyimpan data akun (username, password, peran, dan saldo).
- File produk.csv digunakan untuk menyimpan data produk (ID,nama,harga,stok).Dengan library ini, data dapat disimpan secara permanen tanpa memerlukan database tambahan seperti MySQL.

2. Library prettytable

Library ini digunakan untuk menampilkan data produk dalam bentuk tabel agar lebih rapi dan mudah dibaca di terminal. Misalnya saat melihat daftar produk, data akan tampil dalam format tabel dengan kolom ID, Nama Produk, Harga, dan Stok.

3. Library pwinput

Library ini digunakan untuk menyembunyikan karakter password ketika pengguna mengetikkan kata sandi saat login atau registrasi. Jadi, input password tidak terlihat langsung di layar demi keamanan.

4. Fitur Penanganan Error try dan except

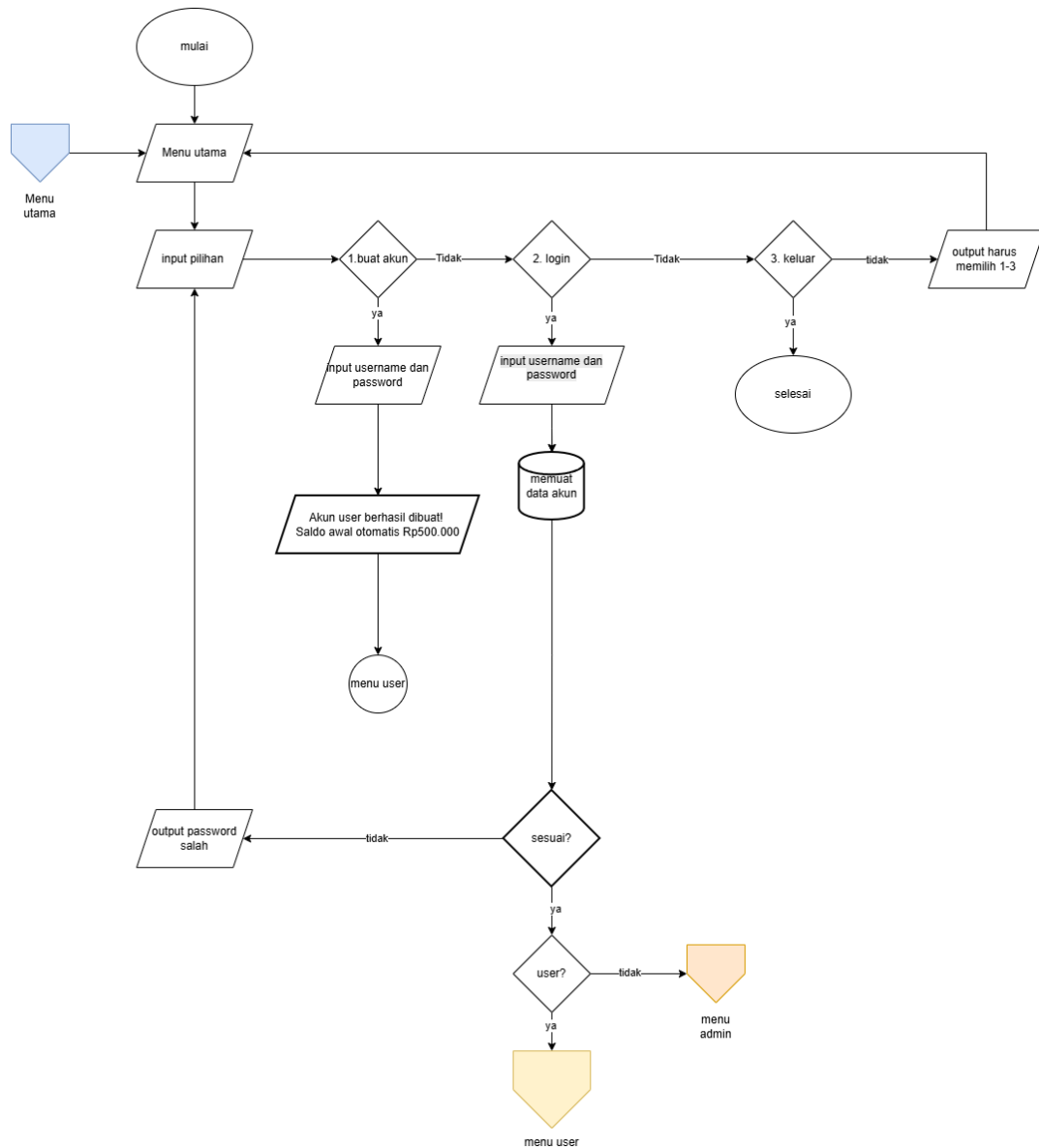
Program dilengkapi dengan error handling menggunakan blok try-except agar tidak berhenti secara tiba-tiba ketika pengguna melakukan kesalahan input. Salah satu error yang ditangani Adalah KeyboardInterrupt, yaitu kesalahan yang terjadi saat pengguna menekan Ctrl + C atau Ctrl + Z di beberapa sistem. Jika hal itu terjadi, program tidak akan crash, melainkan menampilkan pesan seperti:

“Dibatalkan.” atau “Program dihentikan.”

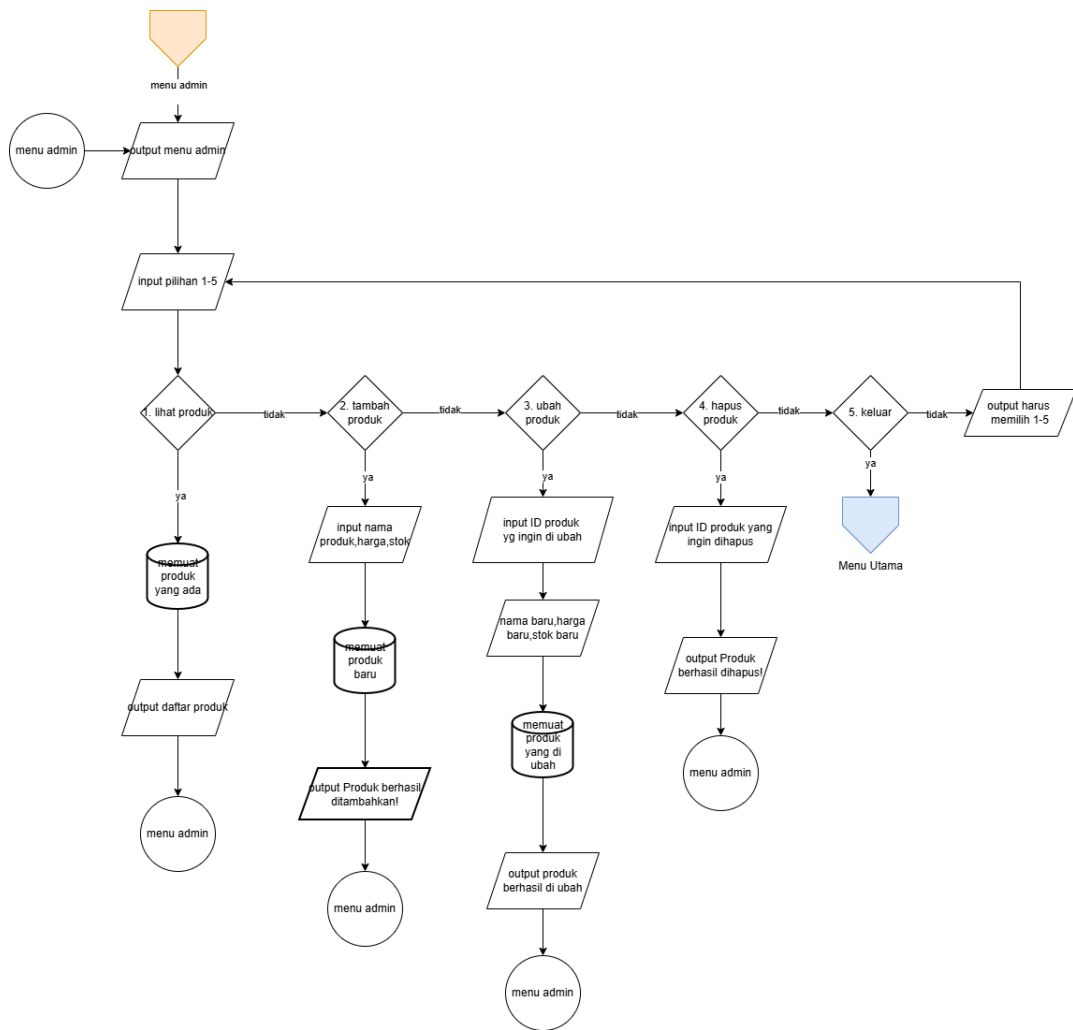
5. Struktur Program dan Fitur Utama

- **Login dan Registrasi:** pengguna dapat membuat akun baru atau login sebagai admin/user.
- **Admin:** dapat melihat, menambah, mengubah, dan menghapus produk.
- **User:** dapat melihat produk, membeli barang, dan melakukan top up saldo.
- **Data Tersimpan Permanen:** semua perubahan tersimpan di file csv
- **Validasi Input:** hanya angka yang valid bisa diterima untuk harga, stok, dan jumlah pembelian.

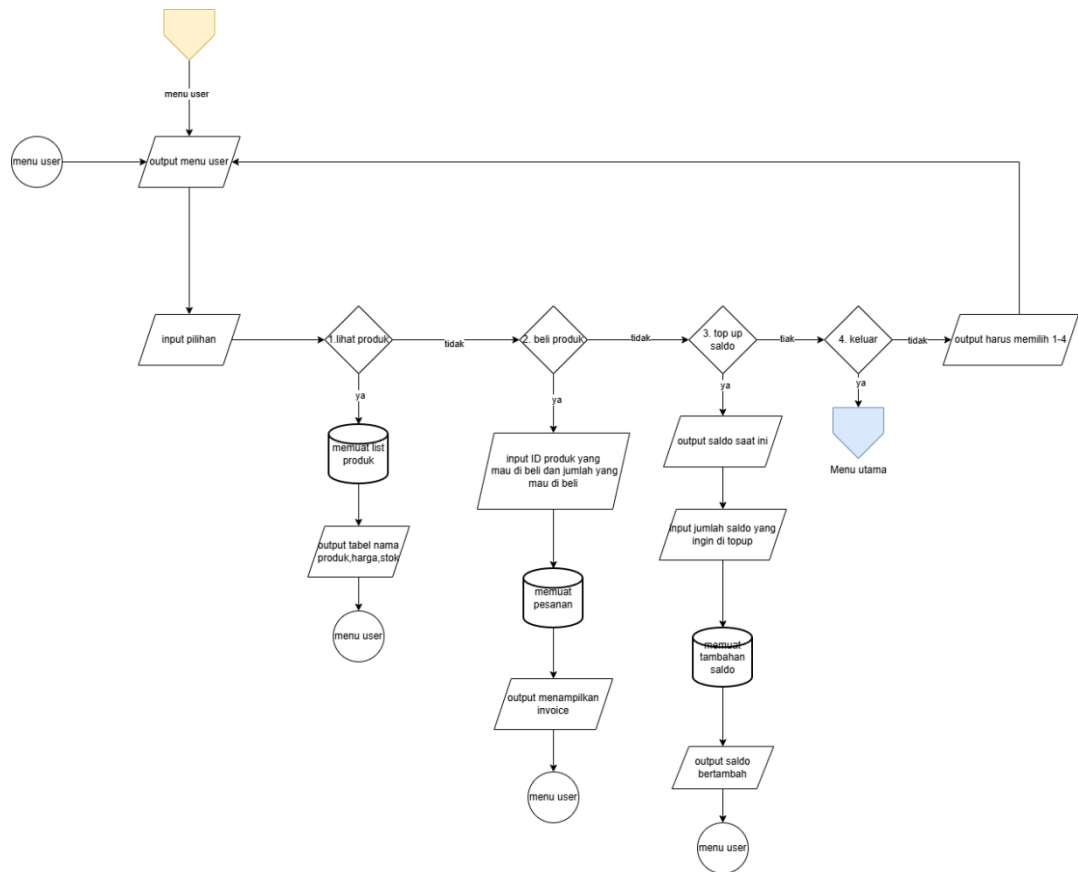
2.2 Flowchart



Gambar 2. 1 Flowchart Menu Utama



Gambar 2. 2 Flowchart Menu admin



Gambar 2. 3 Flowchart Menu User

BAB III

HASIL DAN PEMBAHASAN

3.1 Implementasi Program

Program ini merupakan aplikasi berbasis terminal yang dirancang untuk membantu proses penjualan merchandise komunitas secara digital. Tujuan utama dari aplikasi ini adalah mempermudah pengelolaan data penjualan, baik dari sisi admin maupun pengguna. Dalam implementasinya, aplikasi ini menggunakan bahasa pemrograman Python dan memanfaatkan file CSV sebagai media penyimpanan data agar proses input, pembaruan, dan penghapusan informasi dapat dilakukan secara sederhana tanpa menggunakan database eksternal.

Sistem ini memiliki dua peran utama yaitu admin dan user, di mana keduanya memiliki akses dan fungsi yang berbeda sesuai kebutuhan. Pengguna dengan peran admin dapat menambahkan produk baru ke dalam daftar penjualan, mengubah data produk yang sudah ada, dan menghapus produk yang tidak lagi dijual. Semua data tersebut tersimpan secara otomatis di dalam file produk.csv. Dengan adanya fitur ini, admin dapat lebih mudah memperbarui daftar merchandise yang tersedia tanpa harus mengedit file secara manual.

Sementara itu, pengguna dengan peran user memiliki akses untuk melihat daftar produk yang dijual, membeli produk sesuai keinginan, dan melakukan pengisian saldo (top-up) jika saldo yang dimiliki tidak mencukupi untuk transaksi. Setiap akun user yang baru dibuat secara otomatis akan mendapatkan saldo awal sebesar Rp500.000, yang nantinya akan digunakan untuk melakukan pembelian. Dalam proses pembelian, sistem akan melakukan pengecekan terhadap saldo dan stok barang. Jika saldo tidak mencukupi atau stok produk habis, maka transaksi tidak dapat dilakukan dan sistem akan memberikan notifikasi berupa pesan “Saldo tidak cukup” atau “Stok tidak tersedia.”

Proses *login* dalam aplikasi ini juga dibuat sederhana namun aman. *Admin* maupun *user* harus memasukkan *username* dan *password* yang *valid* untuk dapat mengakses sistem. Aplikasi menggunakan *library* *pwininput* agar *password* yang

dimasukkan tidak terlihat di layar, sehingga memberikan keamanan tambahan. Selain itu, terdapat validasi untuk memastikan data yang diinput benar dan tidak kosong, misalnya harga produk tidak boleh bernilai negatif dan stok harus berupa angka.

Salah satu fitur penting dalam program ini adalah transaksi pembelian produk, yang diimplementasikan melalui fungsi transaksi(user). Ketika pengguna melakukan pembelian, sistem akan secara otomatis menghitung total harga berdasarkan jumlah produk yang dibeli, mengurangi saldo pengguna sesuai total harga, serta memperbarui stok barang di dalam file produk.csv. Setelah transaksi selesai, program akan menampilkan invoice pembelian yang berisi rincian seperti nama pembeli, nama produk, jumlah yang dibeli, total harga, dan sisa saldo. Fitur ini membantu pengguna untuk mengetahui detail pembelian mereka secara transparan.

Selain itu, pengguna juga dapat melakukan pengisian saldo (top-up) melalui fungsi tambahan yang memastikan saldo tidak melebihi batas maksimum, yaitu Rp1.000.000. Dengan adanya pembatasan ini, sistem dapat mencegah kesalahan input dan menjaga stabilitas data dalam aplikasi.

Secara keseluruhan, implementasi program ini mencerminkan proses bisnis sederhana dalam penjualan merchandise komunitas. Mulai dari pengelolaan akun, pengaturan produk, hingga transaksi dan pencatatan saldo, semuanya dibuat agar mudah digunakan dan dapat dijalankan sepenuhnya melalui terminal. Dengan antarmuka yang sederhana namun fungsional, aplikasi ini diharapkan dapat menjadi solusi praktis bagi komunitas dalam mengelola penjualan merchandise mereka dengan cara yang lebih efisien dan terorganisir.

3.2 Alur Program

Program ini merupakan simulasi sistem penjualan merchandise komunitas yang berbasis Python dan berjalan melalui terminal. Program ini dirancang untuk mempermudah proses jual beli sederhana antara pengguna (user) dan pengelola (admin) dalam satu sistem yang terorganisir. Di dalamnya terdapat beberapa fitur utama, seperti pembuatan akun pengguna baru, login berdasarkan peran (admin atau user), pengelolaan data produk oleh admin, pembelian produk oleh user, serta pencetakan invoice transaksi.

Program dimulai dari tampilan menu utama yang menampilkan tiga pilihan yaitu Buat Akun User, Login, dan Keluar. Pada tahap ini, sistem terlebih dahulu memeriksa apakah sudah terdapat akun admin. Jika belum ada, maka program secara otomatis membuat akun admin default dengan username “admin” dan password “123”. Pengguna baru dapat memilih opsi Buat Akun User untuk mendaftar dengan memasukkan username dan password. Setelah berhasil, akun pengguna otomatis mendapat saldo awal sebesar Rp500.000 yang tersimpan di dalam file user.csv.

Setelah akun dibuat, pengguna dapat memilih menu Login. Sistem akan memeriksa kesesuaian data username dan password berdasarkan isi file user.csv. Apabila login berhasil, pengguna diarahkan ke menu sesuai perannya masing-masing. Pengguna dengan peran admin memiliki akses untuk menambah produk baru, melihat daftar produk yang tersedia, mengubah data produk, serta menghapus produk yang sudah tidak dijual. Semua data produk tersimpan dalam file produk.csv, sehingga setiap perubahan akan tersimpan secara permanen.

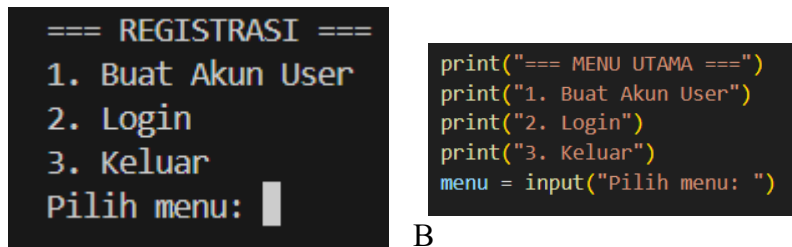
Sementara itu, pengguna dengan peran user memiliki akses untuk melihat produk yang tersedia, membeli produk sesuai kebutuhan, dan melakukan top-up saldo dengan batas maksimal Rp1.000.000. Saat melakukan pembelian, sistem akan menampilkan daftar produk yang ada, meminta pengguna memilih produk dan jumlah yang ingin dibeli, lalu memeriksa apakah stok mencukupi serta saldo pengguna cukup. Jika saldo tidak cukup, sistem akan menampilkan pesan “Saldo tidak cukup”, dan transaksi dibatalkan. Jika transaksi berhasil, maka stok produk

otomatis berkurang, saldo pengguna diperbarui, dan sistem menampilkan invoice transaksi yang berisi nama pembeli, produk yang dibeli, jumlah, total harga, serta sisa saldo pengguna.

Secara keseluruhan, alur kerja program ini dibuat agar mudah dipahami dan efisien untuk dijalankan. Semua aktivitas — mulai dari pembuatan akun, login, pengelolaan produk, transaksi, hingga pembaruan saldo — disimpan ke dalam file CSV agar data tetap tersimpan meskipun program ditutup. Sistem ini sekaligus menjadi simulasi nyata penerapan konsep CRUD (Create, Read, Update, Delete) dalam dunia pemrograman Python untuk mengelola data dan transaksi secara terstruktur.

Berikut Alur program:

1) Menu Utama



B

Gambar 3. 1 Alur program Menu Utama

Pada Menu utama terdapat 3 pilihan yaitu buat akun user, login, dan keluar. Pada pilihan pertama ada buat akun user di situ pengguna akan memasukkan username dan password mereka untuk memasukkan akun mereka ke data dan untuk pengguna pertama akan mendapatkan saldo senilai Rp.500.000. Pilihan ke 2 pengguna dan admin memasukkan username dan password mereka sesuai username dan password di akun mereka. Pilihan ke 3 maka akan keluar dari program.

2) Buat akun user

```
=== REGISTRASI ===
1. Buat Akun User
2. Login
3. Keluar
Pilih menu: 1
=== BUAT AKUN USER ===
Masukkan username : syahla
Masukkan password : ***
Akun user berhasil dibuat! Saldo awal otomatis Rp500.000
=== REGISTRASI ===
1. Buat Akun User
2. Login
3. Keluar
Pilih menu: █
```

```
def buat_akun():
    try:
        print("=== BUAT AKUN USER ===")
        username = input("Masukkan username : ")
        password = pwinput.pwinput("Masukkan password : ")
        saldo_awal = 500000
        with open(database_user, "a", newline="") as file:
            tulis = csv.writer(file)
            tulis.writerow([username, password, "user", saldo_awal])
        print("Akun user berhasil dibuat! Saldo awal Rp500.000")
    except KeyboardInterrupt:
        print("Dibatalkan.")
```

Gambar 3. 2 buat akun user

Pada pilihan 1 memasukkan username dan password sebagai user dan mendapatkan saldo awal Rp.500.000.

3) Login user

```
=== REGISTRASI ===
1. Buat Akun User
2. Login
3. Keluar
Pilih menu: 2
=== LOGIN ===
Masukkan username : syahla
Masukkan password : ***
Login berhasil!
=== MENU USER ===
1. Lihat Produk
2. Beli Produk
3. Top Up Saldo
4. Logout
Pilih menu: █
```

```
def login():
    try:
        print("=== LOGIN ===")
        username = input("Masukkan username : ")
        password = pwinput.pwinput("Masukkan password : ")
        with open(database_user, "r") as file:
            baca = csv.reader(file)
            for baris in baca:
                if baris[0] == username and baris[1] == password:
                    print("Login berhasil!")
                    return baris
            print("Username atau password salah!")
            return None
    except KeyboardInterrupt:
        print("Login dibatalkan.")
        return None
```

Gambar 3. 3 login user

Pada pilihan ke 2 login dan memasukkan username dan password yang sesuai saat membuat akun user sebelumnya dan akan masuk ke menu user.

4) Menu user

```
==== MENU USER ====
1. Lihat Produk
2. Beli Produk
3. Top Up Saldo
4. Logout
Pilih menu: 1
==== DAFTAR PRODUK ====
+---+-----+-----+-----+
| ID | Nama Produk | Harga | Stok |
+---+-----+-----+-----+
| 4  | gacın      | Rp25000 | 50  |
+---+-----+-----+-----+
```

```
def tampilkan_produk():
    try:
        print("==== DAFTAR PRODUK ====")
        with open(database_produk, "r") as file:
            baca = csv.reader(file)
            data = [baris for baris in baca if len(baris) == 4]
            if data:
                tabel = PrettyTable(["ID", "Nama Produk", "Harga", "Stok"])
                for baris in data:
                    tabel.add_row([baris[0], baris[1], f"Rp{baris[2]}", baris[3]])
                print(tabel)
            else:
                print("Belum ada produk.")
    except FileNotFoundError:
        print("Belum ada produk.")
    except KeyboardInterrupt:
        print("Kembali ke menu.")
```

Gambar 3. 4 Menu User

Pilihan pertama di menu user Adalah melihat produk dan dari program akan menampilkan daftar produk yang tersedia.

5) Pembelian Produk

```
==== MENU USER ====
1. Lihat Produk
2. Beli Produk
3. Top Up Saldo
4. Logout
Pilih menu: 2
==== PEMBELIAN PRODUK ====
==== DAFTAR PRODUK ====
+---+-----+-----+-----+
| ID | Nama Produk | Harga | Stok |
+---+-----+-----+-----+
| 4  | gacın      | Rp25000 | 50  |
+---+-----+-----+-----+
Masukkan ID produk yang ingin dibeli: 4
Masukkan jumlah yang ingin dibeli: 10

==== INVOICE ====
Nama Pembeli : cala
Produk       : gacın
Jumlah       : 10
Total Bayar  : Rp 250000
Sisa Saldo   : Rp 250000
```

```
def transaksi(user):
    try:
        print("==== PEMBELIAN PRODUK ====")
        tampilkan_produk()
        id_beli = input("Masukkan ID produk yang ingin dibeli: ")
        jumlah = input("Masukkan jumlah yang ingin dibeli: ")
```

Gambar 3. 5 beli produk

Pilihan kedua yaitu user membeli produk setelah melihat di daftar produk user di minta untuk memasukan nama produk yang ingin idbeli dan jumlah yang ingin di beli, setelah selesai meng input semuanya maka akan menampilkan invoice nya dan sisa saldo yang di miliki user.

6) Top up saldo

```
=== MENU USER ===
1. Lihat Produk
2. Beli Produk
3. Top Up Saldo
4. Logout
Pilih menu: 3
=== TOP UP SALDO ===
Saldo saat ini: Rp430000
Masukkan jumlah saldo yang ingin ditambahkan: 50000
Top up berhasil! Saldo baru Anda: Rp480000
```

```
def topup_saldo(user):
    try:
        print("=== TOP UP SALDO ===")
        saldo = int(user[3])
        print("Saldo saat ini: Rp", saldo)
        tambah = input("Masukkan jumlah saldo yang ingin ditambahkan: ")

        if tambah.isdigit() and int(tambah) > 0:
            tambah = int(tambah)
            if saldo + tambah > 1000000:
                print("Gagal! Saldo maksimal Rp1.000.000")
            else:
                saldo_baru = saldo + tambah
                ubah_saldo(user[0], saldo_baru)
                user[3] = str(saldo_baru)
                print("Top up berhasil! Saldo baru: Rp", saldo_baru)
        else:
            print("Masukkan angka yang sesuai.")
    except KeyboardInterrupt:
        print("Top up dibatalkan.")
```

Gambar 3. 6 Top up saldo

Pada pilihan ketiga user ingin menambahkan saldo, maka user memasukkan jumlah saldo yang ingin di tambahkan dan saldo akan bertambah langsung.

7) logout

```
=== MENU USER ===
1. Lihat Produk
2. Beli Produk
3. Top Up Saldo
4. Logout
Pilih menu: 4
=== REGISTRASI ===
1. Buat Akun User
2. Login
3. Keluar
Pilih menu: █
```

```
elif pilih == "4":
    break
```

Gambar 3. 7 logout

Pada pilihan terakhir ada keluar dari akun user dan program akan Kembali ke menu utama.

8) Menu admin

```
=== REGISTRASI ===
1. Buat Akun User
2. Login
3. Keluar
Pilih menu: 2
=== LOGIN ===
Masukkan username : admin
Masukkan password : ***
Login berhasil!
=== MENU ADMIN ===
1. Lihat Produk
2. Tambah Produk
3. Ubah Produk
4. Hapus Produk
5. Logout
Pilih menu: █
```

```
def cek_admin():
    try:
        with open(database_user, "r") as file:
            baca = csv.reader(file)
            for baris in baca:
                if baris[0] == "admin":
                    return
    except FileNotFoundError:
        pass
    with open(database_user, "w", newline="") as file:
        tulis = csv.writer(file)
        tulis.writerow(["admin", "123", "admin", 0])
```

Gambar 3. 8 menu admin

Saat ingin role admin sama seperti user yaitu login masukkan username dan password setelah login berhasil maka akan masuk ke menu admin.

9) Lihat Produk

```
=== MENU ADMIN ===
1. Lihat Produk
2. Tambah Produk
3. Ubah Produk
4. Hapus Produk
5. Logout
Pilih menu: 1
=== DAFTAR PRODUK ===
+---+-----+-----+-----+
| ID | Nama Produk | Harga | Stok |
+---+-----+-----+-----+
| 3  | kaos       | Rp80000 | 11  |
+---+-----+-----+-----+
```

```
def tampilkan_produk():
    try:
        print("=== DAFTAR PRODUK ===")
        with open(database_produk, "r") as file:
            baca = csv.reader(file)
            data = [baris for baris in baca if len(baris) == 4]
            if data:
                tabel = PrettyTable(["ID", "Nama Produk", "Harga", "Stok"])
                for baris in data:
                    tabel.add_row([baris[0], baris[1], f"Rp{baris[2]}", baris[3]])
                print(tabel)
            else:
                print("Belum ada produk.")
    except FileNotFoundError:
        print("Belum ada produk.")
    except KeyboardInterrupt:
        print("Kembali ke menu.")
```

Gambar 3. 9 lihat produk

10) Tambah produk

```
=== MENU ADMIN ===
1. Lihat Produk
2. Tambah Produk
3. Ubah Produk
4. Hapus Produk
5. Logout
Pilih menu: 2
=== TAMBAH PRODUK ===
Nama produk : gacin
Harga produk : 25000
Stok produk : 50
Produk ID 4 berhasil ditambahkan!
```

```
# Tambah produk
Windsurf: Refactor | Explain | Generate Docstring | X
def tambah_produk():
    try:
        print("=== TAMBAH PRODUK ===")
        nama = input("Nama produk : ")
        harga = input("Harga produk : ")
        stok = input("Stok produk : ")

        if harga.isdigit() and stok.isdigit() and int(harga) > 0 and int(stok) > 0:
            with open(database_produk, "r") as file:
                data = list(csv.reader(file))
                id_baru = len(data) + 1

            with open(database_produk, "a", newline="") as file:
                tulis = csv.writer(file)
                tulis.writerow([id_baru, nama, harga, stok])
            print(f"Produk ID {id_baru} berhasil ditambahkan!")
        else:
            print("Masukkan angka yang sesuai.")
    except KeyboardInterrupt:
        print("Dibatalkan.")
```

Gambar 3. 10 Tambah Produk

Pada pilihan ke 2 admin ingin menambahkan produk maka akan menginput nama produk nya, harga produk tersebut, dan stok nya berapa.

11) ubah produk

```
=== MENU ADMIN ===
1. Lihat Produk
2. Tambah Produk
3. Ubah Produk
4. Hapus Produk
5. Logout
Pilih menu: 3
=== UBAH PRODUK ===
=== DAFTAR PRODUK ===
+-----+-----+-----+
| ID | Nama Produk | Harga | Stok |
+-----+-----+-----+
| 3 | kaos | Rp80000 | 11 |
| 4 | gacin | Rp25000 | 50 |
+-----+-----+-----+
Masukkan ID produk yang ingin diubah: 3
Nama baru : tshirt
Harga baru : 100000
Stok baru : 45
Produk berhasil diubah!
```

```
def ubah_produk():
    try:
        print("=== UBAH PRODUK ===")
        tampilkan_produk()
        id_produk = input("Masukkan ID produk yang ingin diubah: ")

        semua_data = []
        ditemukan = False

        with open(database_produk, "r") as file:
            baca = csv.reader(file)
            for baris in baca:
                if baris[0] == id_produk:
                    nama = input("Nama baru : ")
                    harga = input("Harga baru : ")
                    stok = input("Stok baru : ")
                    if harga.isdigit() and stok.isdigit():
                        semua_data.append([baris[0], nama, harga, stok])
                        ditemukan = True
                    else:
                        print("Masukkan angka yang sesuai.")
                        return
                else:
                    semua_data.append(baris)
```

Gambar 3. 11 Tambah Produk

Pada pilihan ke 3 yaitu meng ubah produk jadi disini admin disuru memasukkan ID produk yang ingin di hapus.

12) Hapus produk

```

===== MENU ADMIN =====
1. Lihat Produk
2. Tambah Produk
3. Ubah Produk
4. Hapus Produk
5. Logout
Pilih menu: 4
===== HAPUS PRODUK =====
===== DAFTAR PRODUK =====

+-----+-----+-----+-----+
| ID | Nama Produk | Harga | Stok |
+-----+-----+-----+-----+
| 3 | tshirt | Rp100000 | 45 |
| 4 | gacin | Rp25000 | 50 |
+-----+-----+-----+-----+
Masukkan ID produk yang ingin dihapus: 3
Produk berhasil dihapus!

```

Gambar 3. 12 hapus produk

Pada pilihan ke 4 yaitu menghapus produk admin di minta untuk menginput ID produk yang ingin di hapus.

13) Logout

```

=== MENU ADMIN ===
1. Lihat Produk
2. Tambah Produk
3. Ubah Produk
4. Hapus Produk
5. Logout
Pilih menu: 5
=== MENU UTAMA ===
1. Buat Akun User
2. Login
3. Keluar
Pilih menu:

```

```

elif pilih == "5":
    break

```

Gambar 3. 13 logout

Pada pilihan terakhir yaitu logout dari menu admin

3.3 Source Code

Tabel 3. 1 Source Code

```
import csv
from prettytable import PrettyTable
import pwinput

database_user = "user.csv"
database_produk = "produk.csv"

# cek admin
def cek_admin():
    try:
        with open(database_user, "r") as file:
            baca = csv.reader(file)
            for baris in baca:
                if baris[0] == "admin":
                    return
    except FileNotFoundError:
        pass
    with open(database_user, "w", newline="") as file:
        tulis = csv.writer(file)
        tulis.writerow(["admin", "123", "admin",
0])
```

Tabel 3. 2 Source Code

```
# cek produk
def cek_produk():
    try:
        with open(database_produk, "r") as file:
            pass
    except FileNotFoundError:
        with open(database_produk, "w",
newline="") as file:
            pass

# buat akun
def buat_akun():
    try:
        print("=== BUAT AKUN USER ===")
        username = input("Masukkan username : ")
        if username.strip() == "":
            print("Username          tidak          boleh
kosong!")
            return
        password = pwininput.pwininput("Masukkan
password : ")
        if password.strip() == "":
            print("Username          tidak          boleh
kosong!")
            return
        saldo_awal = 500000
        with open(database_user, "a", newline="")
as file:
            tulis = csv.writer(file)
            tulis.writerow([username, password,
"user", saldo_awal])
            print("Akun user berhasil dibuat! Saldo
awal Rp500.000")
```

Tabel 3. 3 Source Code

```
except
KeyboardInterrupt:
    print("Dibatalkan.")

# login
def login():
    try:
        print("=== LOGIN ===")
        username = input("Masukkan username : ")
        password = pwininput("Masukkan
password : ")
        with open(database_user, "r") as file:
            baca = csv.reader(file)
            for baris in baca:
                if baris[0] == username and
baris[1] == password:
                    print("Login berhasil!")
                    return baris
            print("Username atau password salah!")
            return None
    except KeyboardInterrupt:
        print("Login dibatalkan.")
        return None
```

Tabel 3. 4 Source Code

```
# tambah produk
def tambah_produk():
    try:
        print("=== TAMBAH PRODUK ===")
        nama = input("Nama produk : ")
        harga = input("Harga produk : ")
        stok = input("Stok produk : ")

        if harga.isdigit() and stok.isdigit() and
int(harga) > 0 and int(stok) > 0:
            with open(database_produk, "r") as
file:
                data = list(csv.reader(file))
                id_baru = len(data) + 1

                with open(database_produk, "a",
newline="") as file:
                    tulis = csv.writer(file)
                    tulis.writerow([id_baru, nama,
harga, stok])
                    print(f"Produk ID {id_baru} berhasil
ditambahkan!")
            else:
                print("Masukkan angka yang sesuai.")
    except KeyboardInterrupt:
        print("Dibatalkan.")

# tampilkan produk
def tampilkan_produk():
    try:
```

Tabel 3. 5 Source Code

```
print("===          DAFTAR
PRODUK ===")

    with open(database_produk, "r") as file:
        baca = csv.reader(file)
        data = [baris for baris in baca if
len(baris) == 4]
        if data:
            tabel = PrettyTable(["ID", "Nama
Produk", "Harga", "Stok"])
            for baris in data:
                tabel.add_row([baris[0],
baris[1], f"Rp{baris[2]}", baris[3]])
            print(tabel)
        else:
            print("Belum ada produk.")
    except FileNotFoundError:
        print("Belum ada produk.")
    except KeyboardInterrupt:
        print("Kembali ke menu.")

# ubah produk
def ubah_produk():
    try:
        print("=== UBAH PRODUK ===")
        tampilkan_produk()
        id_produk = input("Masukkan ID produk
yang ingin diubah: ")

        semua_data = []
        ditemukan = False
```

Tabel 3. 6 Source Code

```

                                with open(database_produk,
"r") as file:
    baca = csv.reader(file)
    for baris in baca:
        if baris[0] == id_produk:
            nama = input("Nama baru : ")
            harga = input("Harga baru : ")
            stok = input("Stok baru : ")
                        if harga.isdigit() and
stok.isdigit():
                                semua_data.append([baris[0],
nama, harga, stok])
                                ditemukan = True
                                else:
                                    print("Masukkan angka yang
sesuai.")
                                    return
                                else:
                                    semua_data.append(baris)

        if ditemukan:
            with open(database_produk, "w",
newline="") as file:
                tulis = csv.writer(file)
                tulis.writerows(semua_data)
                print("Produk berhasil diubah!")
            else:
                print("Produk tidak ditemukan.")
    except KeyboardInterrupt:
        print("Dibatalakan.")

# hapus produk
def hapus_produk():
    try:
        print("=== HAPUS PRODUK ===")
        tampilkan_produk()

```

Tabel 3. 7 Source Code

```

                                id_produk      =
input("Masukkan ID produk yang ingin dihapus: ")

    semua_data = []
    ditemukan = False

    with open(database_produk, "r") as file:
        baca = csv.reader(file)
        for baris in baca:
            if baris[0] == id_produk:
                ditemukan = True
            else:
                semua_data.append(baris)

    if ditemukan:
        with open(database_produk, "w",
newline="") as file:
            tulis = csv.writer(file)
            tulis.writerows(semua_data)
            print("Produk berhasil dihapus!")
    else:
        print("Produk tidak ditemukan.")
except KeyboardInterrupt:
    print("Dibatalkan.")

# transaksi
def transaksi(user):
    try:
        print("=== PEMBELIAN PRODUK ===")
        tampilkan_produk()
        id_beli = input("Masukkan ID produk yang
ingin dibeli: ")
        jumlah = input("Masukkan jumlah yang
ingin dibeli: ")

```

Tabel 3. 8 Source Code

```
        if id_beli.isdigit()
and jumlah.isdigit() and int(id_beli) > 0 and
int(jumlah) > 0:
            semua_data = []
            produk_ada = False
            jumlah = int(jumlah)

            with open(database_produk, "r") as
file:
                baca = csv.reader(file)
                for baris in baca:
                    if baris[0] == id_beli:
                        produk_ada = True
                        harga = int(baris[2])
                        stok = int(baris[3])
                        total = harga * jumlah

                        if jumlah > stok:
                            print("Stok tidak
cukup!")

                            return

                        saldo = int(user[3])
                        if total > saldo:
                            print("Saldo tidak
cukup!")

                            return
```


Tabel 3. 9 Source Code

```
baris[3] = str(stok - jumlah)
        semua_data.append(baris)
        saldo_baru = saldo - total
        ubah_saldo(user[0],
saldo_baru)
        user[3] = str(saldo_baru)

        print("\n=== INVOICE ===")
        print("Nama Pembeli :",
user[0])
        print("Produk      :",
baris[1])
        print("Jumlah      :",
jumlah)
        print("Total Bayar   :
Rp", total)
        print("Sisa Saldo    :
Rp", saldo_baru)
    else:
        semua_data.append(baris)

        if produk_ada:
            with open(database_produk, "w",
newline="") as file:
                tulis = csv.writer(file)
                tulis.writerows(semua_data)
        else:
            print("Produk tidak ditemukan.")
    else:
        print("Masukkan angka yang sesuai.")
except KeyboardInterrupt:
    print("Transaksi dibatalkan.")

# top up saldo
def topup_saldo(user):
    try:
```

Tabel 3. 10 Source Code

```
print("=== TOP UP SALDO  
===")  
  
    saldo = int(user[3])  
    print("Saldo saat ini: Rp", saldo)  
    tambah = input("Masukkan jumlah saldo  
yang ingin ditambahkan: ")  
  
    if tambah.isdigit() and int(tambah) > 0:  
        tambah = int(tambah)  
        if saldo + tambah > 1000000:  
            print("Gagal! Saldo maksimal  
Rp1.000.000")  
        else:  
            saldo_baru = saldo + tambah  
            ubah_saldo(user[0], saldo_baru)  
            user[3] = str(saldo_baru)  
            print("Top up berhasil! Saldo  
baru: Rp", saldo_baru)  
        else:  
            print("Masukkan angka yang sesuai.")  
    except KeyboardInterrupt:  
        print("Top up dibatalkan.")  
  
# ubah saldo  
def ubah_saldo(nama_user, saldo_baru):  
    semua_user = []  
    with open(database_user, "r") as file:  
        baca = csv.reader(file)  
        for baris in baca:
```

Tabel 3. 11 Source Code

```
if baris[0] == nama_user:
    baris[3] = str(saldo_baru)
    semua_user.append(baris)
    with open(database_user, "w", newline="") as
file:
        tulis = csv.writer(file)
        tulis.writerows(semua_user)

# menu utama
cek_admin()
cek_produk()

while True:
    try:
        print("=== MENU UTAMA ===")
        print("1. Buat Akun User")
        print("2. Login")
        print("3. Keluar")
        menu = input("Pilih menu: ")

        if menu == "1":
            buat_akun()
        elif menu == "2":
            user = login()
            if user:
                if user[2] == "admin":
                    while True:
                        print("=== MENU ADMIN
===")
```

Tabel 3. 12 Source Code

```
print("1. Lihat Produk")
        print("2. Tambah Produk")
        print("3. Ubah Produk")
        print("4. Hapus Produk")
        print("5. Logout")
        pilih = input("Pilih
menu: ")

        if pilih == "1":
            tampilkan_produk()
        elif pilih == "2":
            tambah_produk()
        elif pilih == "3":
            ubah_produk()
        elif pilih == "4":
            hapus_produk()
        elif pilih == "5":
            break
        else:
            print("Pilihan tidak
sesuai.")

        elif user[2] == "user":
            while True:
print("=== MENU USER ===")
                print("1. Lihat Produk")
print("2. Beli Produk")
print("3. Top Up Saldo")
                print("4. Logout")
pilih = input("Pilih menu: ")
                if pilih == "1":
tampilkan_produk()
                elif pilih == "2":
transaksi(user)
                elif pilih == "3":
topup_saldo(user)
```

Tabel 3. 13 Source Code

```
elif pilih == "4":  
    break  
    else:  
        print("Pilihan tidak  
sesuai.")  
        elif menu == "3":  
            print("Terima kasih telah menggunakan  
program!")  
            break  
        else:  
            print("Pilihan tidak sesuai.")  
except (KeyboardInterrupt, EOFError):  
    print("Program dihentikan.")  
    break
```

BAB IV

PENUTUP

4.1 Kesimpulan

Dari hasil pengembangan aplikasi penjualan merchandise komunitas berbasis Python, dapat disimpulkan bahwa aplikasi ini berhasil memberikan solusi sederhana namun efektif dalam membantu proses pengelolaan data produk, transaksi, dan pengguna. Sistem ini mampu menangani berbagai fungsi utama, seperti pembuatan akun pengguna, login sesuai peran, pengelolaan produk oleh admin, serta proses pembelian dan pengurangan saldo oleh user. Dengan menggunakan file CSV sebagai media penyimpanan, aplikasi ini dapat menyimpan data secara terstruktur dan mudah diakses kembali tanpa perlu basis data kompleks. Aplikasi ini juga menerapkan konsep dasar pemrograman seperti percabangan, perulangan, fungsi, dan manipulasi file untuk mengatur alur logika program. Meskipun berbasis teks dan dijalankan di terminal, sistem ini sudah mencerminkan prinsip dasar sistem informasi sederhana yang efisien dan mudah dipahami, serta dapat digunakan untuk mempermudah kegiatan penjualan dalam skala kecil maupun menengah.

4.2 Saran

Untuk pengembangan lebih lanjut, terdapat beberapa saran agar aplikasi ini dapat menjadi lebih baik dan fungsional, yaitu:

1. Menambahkan antarmuka grafis (GUI) agar tampilan lebih menarik dan mudah digunakan oleh pengguna awam.
2. Mengintegrasikan sistem dengan database seperti MySQL agar pengelolaan data lebih cepat, aman, dan efisien.
3. Menambahkan fitur laporan penjualan otomatis untuk membantu admin memantau transaksi dalam jangka waktu tertentu.
4. Mengembangkan fitur notifikasi atau riwayat transaksi agar pengguna dapat melihat aktivitas pembelian sebelumnya.
5. Meningkatkan keamanan login dan data pengguna, misalnya dengan enkripsi password atau sistem autentikasi ganda.
6. Menambahkan validasi input yang lebih ketat agar kesalahan pengguna dapat diminimalisir saat mengisi data.

DAFTAR PUSTAKA

Nama youtube : Kelas terbuka

<https://youtu.be/ObTWBJ4QCPQ?si=0tqI55TtTg53VtoW>

https://youtu.be/g2Fd8ruSnw4?si=Zqq3b_ChXiiE1N35

LAMPIRAN

Lampiran 1 : Tabel Kontribusi

Nama	Kontribusi	Bagian
Farrel Fazzadinata Sumarna (2509116111)	README	1. Pengembangan README
Dliya Syahla Hariyanto (2509116095)	Coding, Flowchart, Laporan,README	1. Logika program 2. Flowchart program 3. Coding 4. <i>Finishing</i> Laporan 5. Pengembangan README
Riko Azkarazendra Russianto (2509116104)	README	1. Pengembangan README

Tabel 1. 1 Tabel Kontribusi



konsul terakhir