

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



PENGELOLAAN DAN PENJUALAN IKAN HIAS

Oleh:

Kelompok 6

RIDHO SETIAWAN	2409106029
AKHMAD ZIFA AL FATIH	2409106025
FEBRIAN PRATAMA SAPUTRA	2409106033

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

KATA PENGANTAR

Segala puji dan syukur kami panjatkan ke hadirat Allah SWT atas rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan laporan akhir Praktikum Algoritma Pemrograman Dasar ini dengan baik. Laporan ini disusun sebagai bagian dari evaluasi akhir praktikum untuk mendokumentasikan pembelajaran, pengaplikasian algoritma, dan keterampilan pemrograman yang telah dipelajari. Laporan ini juga diharapkan dapat menjadi acuan evaluasi dalam menyelesaikan permasalahan secara sistematis.

Program kami, "Pengelolaan dan Penjualan Ikan Hias," dirancang untuk membantu pengelolaan penjualan di toko ikan hias dengan fitur pencatatan jenis ikan, harga, dan kelangkaannya, serta mendukung proses pembelian oleh pelanggan.

Selama penyusunan program, kami menghadapi kendala seperti pemahaman logika pemrograman yang kompleks, integrasi modul, dan debugging. Manajemen waktu juga menjadi tantangan karena harus membagi fokus antara perkuliahan dan pengembangan program. Semua kendala ini berhasil diatasi berkat dukungan dari asisten praktikum, rekan diskusi, dan kerja sama tim.

Kami berharap laporan ini tidak hanya menjadi dokumentasi pembelajaran, tetapi juga bermanfaat untuk pengembangan kemampuan di masa depan. Terima kasih kami sampaikan kepada semua pihak yang telah memberikan dukungan, baik secara teknis maupun moral, selama proses ini.

Samarinda, 21 November 2024

Kelompok 6

TAKARIR

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

<i>User</i>	Pengguna
<i>Username</i>	Nama Pengguna
<i>Password</i>	Kata Sandi
<i>Input</i>	Masukan
<i>Role</i>	Peran
<i>Login</i>	Masuk
<i>Register</i>	Registrasi
<i>Create</i>	Membuat
<i>Read</i>	Membaca
<i>Update</i>	Memperbarui
<i>Delete</i>	Menghapus
<i>Source Code</i>	Sumber Kode
<i>Flowchart</i>	Bagan alir
<i>List</i>	Daftar
<i>Tuple</i>	Rangkap
<i>Dictionary</i>	Kamus
<i>Error Handling</i>	Penanganan masalah
<i>Arrow Keys</i>	Tombol Tanda Panah
<i>Enter Key</i>	Tombol “Enter”
<i>Exit</i>	Keluar
<i>Checkout</i>	Pembayaran

DAFTAR ISI

KATA PENGANTAR	2
TAKARIR	3
DAFTAR ISI	4
DAFTAR GAMBAR	6
BAB I	
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Kebutuhan Fungsional.....	1
1. Fungsi untuk Admin	1
2. Fungsi untuk User	1
3. Fungsi Tambahan	2
4. Antarmuka Pengguna	2
5. Keamanan dan Otorisasi	3
6. File CSV	3
1.3 Rumusan Masalah	3
1.4 Batasan Masalah.....	4
1.5 Tujuan.....	4
BAB II	
PERANCANGAN	6
2.1 Analisis Program.....	6
2.2 Flowchart	7
2.3 Konsep/Materi Praktikum yang dipakai	9
BAB III	
HASIL DAN PEMBAHASAN	12
3.1 Tampilan Program.....	12
1. Tampilan Menu Utama	12
2. Tampilan Login	12
3. Tampilan Register	13
4. Tampilan Menu User	13
5. Tampilan Daftar Ikan Hias.....	14
6. Tampilan Pesan Ikan Hias	15
7. Tampilan Keranjang Belanja	15

8. Tampilan Checkout	16
9. Tampilan Menu Admin.....	16
10. Tampilan Penambahan Ikan Hias	17
11. Tampilan Pembacaan Ikan Hias	18
12. Tampilan Pembaruan Ikan Hias	19
13. Tampilan Penghapusan Ikan Hias	20
3.2 Source Code.....	20
BAB IV	
PENUTUP	45
4.1 Kesimpulan	45
4.2 Saran.....	45
DAFTAR PUSTAKA	48
LAMPIRAN	49

DAFTAR GAMBAR

Gambar 2. 1	7
Gambar 3. 1	12
Gambar 3. 2	12
Gambar 3. 3	13
Gambar 3. 4	14
Gambar 3. 5	15
Gambar 3. 6	16
Gambar 3. 7	17
Gambar 3. 8	17
Gambar 3. 9	18
Gambar 3. 10	19
Gambar 3. 11	20
Gambar 3. 12	21
Gambar 3. 13	22
Lampiran 1	51

BAB I

PENDAHULUAN

1.1 Latar Belakang

Untuk menyelesaikan masalah dalam penjualan ikan hias, maka dibuat program “pengelolaan dan penjualan ikan hias” guna memudahkan penjual dalam menjual barang dan menyesuaikan harga dan stok dan juga memudahkan akses pembeli dalam membeli ikan hias.

1.2 Kebutuhan Fungsional

1. Fungsi untuk Admin

- **Tambah Ikan Hias Baru**

Admin dapat menambah data ikan hias baru ke dalam sistem dengan memasukkan informasi seperti jenis ikan, kelangkaan, harga, dan stok. Data tersebut kemudian disimpan dalam file CSV.

- **Update Data Ikan Hias**

Admin dapat memperbarui data ikan hias yang sudah ada, termasuk jenis ikan, kelangkaan, harga, dan stok ikan. Admin dapat memilih ikan berdasarkan nomor urut yang ditampilkan.

- **Hapus Ikan Hias**

Admin dapat menghapus data ikan hias dari sistem dengan memilih ikan berdasarkan nomor urut dan menghapusnya dari file CSV.

2. Fungsi untuk User

- **Pesan Ikan Hias**

User dapat melihat daftar ikan hias yang tersedia dan memesan ikan hias berdasarkan nomor urut yang dipilih. Jumlah ikan yang dipesan juga dapat dimasukkan. Data pesanan kemudian disimpan dalam file CSV.

- **Tampilkan Keranjang Belanja**

User dapat melihat ikan-ikan yang telah dipesan dan ada dalam keranjang belanja. User juga dapat memilih untuk menghapus ikan dari keranjang dan stok ikan akan diperbarui sesuai jumlah yang dipesan.

- **Checkout dan Pembayaran**

User dapat melakukan pembayaran setelah memilih ikan hias di keranjang. Program menghitung total harga, dan jika pembayaran mencukupi, transaksi dianggap berhasil. Setelah pembayaran, data keranjang akan dihapus dan stok ikan akan diperbarui.

3. Fungsi Tambahan

- **Validasi Input**

Program memiliki berbagai jenis validasi input untuk memastikan bahwa input yang dimasukkan oleh pengguna valid, seperti validasi untuk input angka dan huruf.

- **Menyimpan dan Membaca Data**

Semua data ikan hias dan keranjang belanja disimpan dalam file CSV untuk memastikan persistensi data diluar memori sementara program berjalan.

4. Antarmuka Pengguna

- **Tampilan Menu Admin dan User**

Menyediakan antarmuka berbasis teks untuk admin dan user, termasuk pilihan untuk menambah, mengubah, atau menghapus ikan hias, serta untuk memesan ikan dan melihat keranjang belanjaan.

- **Loading dan Pesan Konfirmasi**

Ada fitur *loading* yang menunjukkan proses sementara dan konfirmasi saat melakukan tindakan penting (misalnya saat menghapus atau memesan ikan hias).

5. Keamanan dan Otorisasi

- **Peran Admin dan User**

Program membedakan antara admin dan user. Admin memiliki kontrol penuh atas data ikan hias, sedangkan user hanya dapat melihat dan memesan ikan yang tersedia.

6. File CSV

- **Menyimpan Daftar Ikan**

Daftar ikan hias yang tersedia disimpan dalam file **daftar_ikan.csv** dengan kolom jenis ikan, kelangkaan, harga, dan stok.

- **Menyimpan Keranjang Belanja**

Setiap pesanan yang dilakukan user disimpan dalam file **keranjang.csv** yang mencatat jenis ikan, kelangkaan, harga, jumlah, dan username user.

1.3 Rumusan Masalah

Pada sistem manajemen ikan hias ini, terdapat beberapa masalah yang perlu diatasi, antara lain:

1. Bagaimana cara mengelola data ikan hias yang tersedia, termasuk menambah, mengubah, dan menghapus data ikan hias secara efisien?
2. Bagaimana memungkinkan admin untuk memantau stok ikan hias yang ada dan mengontrol kelangkaan serta harga ikan?
3. Bagaimana menyediakan fitur pemesanan ikan hias oleh user dengan cara yang mudah dan aman?
4. Bagaimana memproses transaksi pembayaran secara otomatis dan memperbarui stok ikan setelah pesanan berhasil diproses?
5. Bagaimana menjaga agar data keranjang belanja tetap tercatat dengan baik agar dapat diakses kembali?

1.4 Batasan Masalah

Untuk menjaga fokus dalam pengembangan sistem ini, beberapa batasan masalah ditetapkan sebagai berikut:

1. Sistem hanya mendukung dua jenis pengguna, yaitu admin dan user. Admin memiliki hak untuk mengelola data ikan hias, sedangkan user hanya bisa memesan ikan.
2. Sistem menggunakan file CSV sebagai media penyimpanan data untuk ikan hias dan keranjang belanja.
3. Sistem hanya mencakup manajemen ikan hias yang dapat dipesan oleh user dan tidak mencakup jenis ikan lain atau fitur terkait seperti pengiriman.
4. Pembayaran dilakukan secara manual (tidak ada integrasi dengan sistem pembayaran elektronik atau gateway).
5. Tidak ada otentikasi atau proteksi tambahan pada data pengguna atau admin. Pengguna dapat mengakses seluruh fungsionalitas sesuai peran yang ditentukan dalam sistem.
6. Sistem pada admin tidak mempunyai pencatatan history pembelian user

1.5 Tujuan

Tujuan dari pengembangan sistem ini adalah:

1. **Membantu Admin Mengelola Ikan Hias**
Memberikan kemudahan bagi admin dalam menambah, mengedit, atau menghapus data ikan hias serta memantau stok ikan yang tersedia.
2. **Mempermudah User dalam Memesan Ikan Hias**
Memberikan platform yang mudah digunakan bagi user untuk memilih dan memesan ikan hias yang mereka inginkan.
3. **Proses Transaksi yang Efisien**
Memastikan transaksi pembelian ikan hias dapat dilakukan dengan cara yang mudah dan aman, termasuk perhitungan harga dan pemutakhiran stok ikan.

4. Mengoptimalkan Pengelolaan Stok dan Harga

Membantu admin untuk selalu mengetahui stok ikan yang tersedia dan melakukan penyesuaian harga serta kelangkaan ikan secara real-time.

BAB II

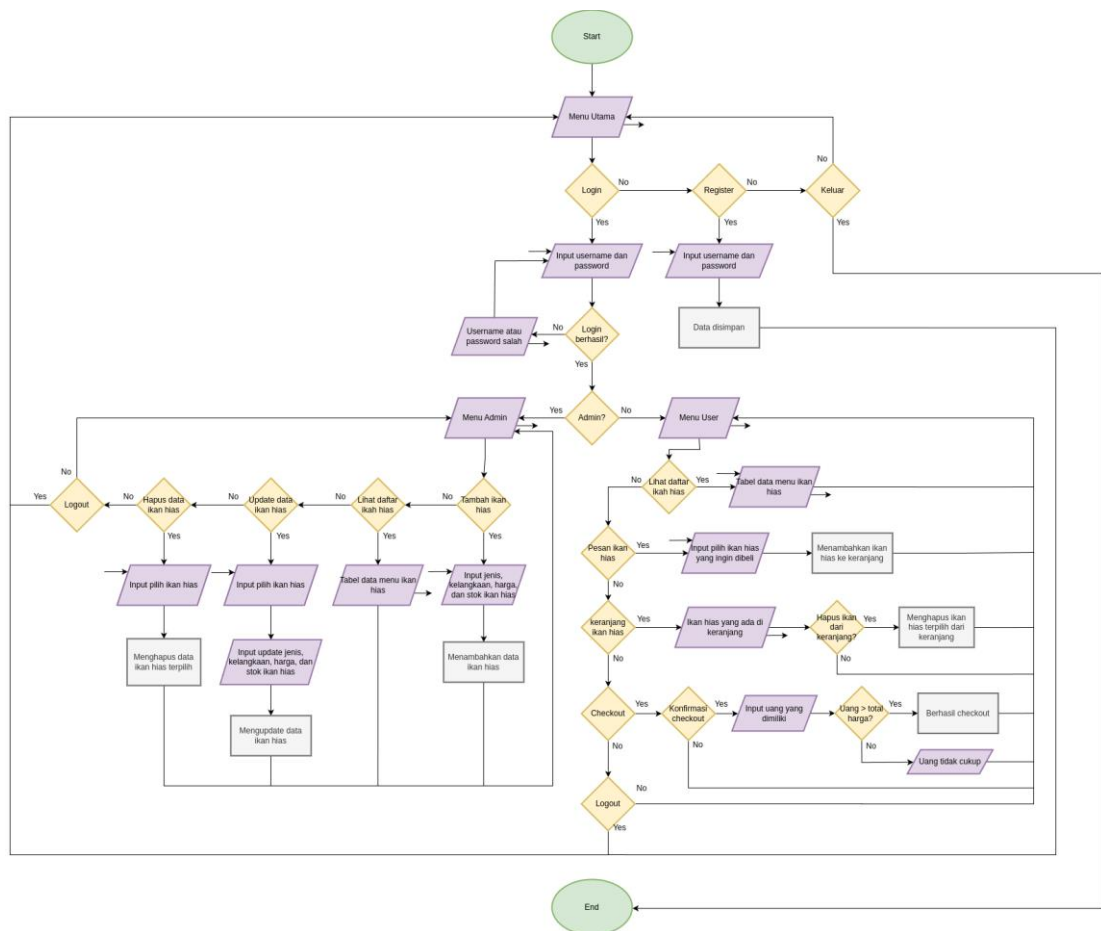
PERANCANGAN

2.1 Analisis Program

Pertama, program akan dimulai dengan menu utama. Di sini pengguna akan dipersilahkan untuk *login* atau *register* sebuah akun atau keluar dari program. Jika *register*, maka pengguna akan membuat sebuah akun baru dengan *username* dan *password*. Jika *login*, maka pengguna akan memasukkan *username* dan *password* dengan akun yang telah dibuat.

Selanjutnya, tergantung *role* akun yang dipakai oleh pengguna, pengguna akan dialihkan ke menu admin atau menu *user*. Jika *role* admin, maka pengguna memiliki akses CRUD (*Create, Read, Update, Delete*) kepada data CSV dengan menu admin. Jika *role user*, maka pengguna akan dialihkan ke menu *user* dimana pengguna bisa membaca data, melakukan pemesanan, melihat keranjang belanja, dan melakukan pembayaran. Kedua menu memiliki pilihan *logout* jika pengguna ingin berpindah akun atau keluar dari program.

2.2 Flowchart



Gambar 2. 1

Pada Flowchart dari yang kami buat, terbagi menjadi 3 bagian yaitu, menu utama, menu admin dan menu user.

1. Menu Utama : dapat diketahui program dapat dimulai dari tampilan menu utama, lalu admin/user diperintahkan untuk memilih login, register atau keluar.
 - a. Login : Jika memilih login maka diperintahkan untuk menginput username dan password, jika sudah teregister maka “login berhasil” dan melanjutkan sesuai role admin atau user, jika username atau password salah maka kembali diperintahkan untuk menginput username dan password kembali.

- b. Register : jika memilih register maka diperintahkan untuk menginput username dan password baru, jika sudah maka data disimpan.
 - c. Logout : Jika memilih keluar maka program berhenti.
- 2. Menu Admin : Pada menu admin ini, admin dapat menambah ikan hias, melihat daftar ikan hias, mengupdate ikan hias, menghapus data ikan hias, dan logout.
 - a. Menambah Ikan Hias : Admin diperintahkan untuk menginput jenis, kelangkaan, harga, dan stok ikan hias, jika sudah menginput maka data akan diproses dan disimpan.
 - b. Melihat Daftar Ikan Hias : Admin dapat melihat output dari data yang tersimpan oleh program.
 - c. Mengupdate Data Ikan Hias : Admin diperintahkan untuk menginput pilihan ikan hias yang ingin di update, lalu yang diupdate yaitu ; update jenis, kelangkaan, harga dan stok ikan hias, jika sudah maka program akan memproses ikan hias yang di update oleh admin.
 - d. Menghapus Data Ikan Hias : Admin diperintahkan untuk menginput pilihan ikan hias yang ingin dihapus, jika sudah maka program akan memproses untuk menghapus data ikan hias yang terpilih.
 - e. Logout : Admin dapat kembali ke menu utama jika memilih logout.
- 3. Menu User : Pada menu user, user dapat melihat daftar ikan hias, memesan ikan hias, memasukan ke keranjang belanja ikan hias, meng *checkout* ikan hias, dan logout.
 - a. Melihat Daftar Ikan Hias : User dapat melihat output dari tabel data menu ikan hias.

- b. Memesan Ikan Hias : User dapat menginput pilihan ikan hias yang ingin dibeli, jika iya maka program akan memproses data ikan yang dipesan dan menambahkan ikan hias ke keranjang.
- c. Memasukan ke Keranjang Belanja Ikan Hias : User dapat melihat output yang diberikan program dari data pesanan ikan hias, jika ingin menghapus ikan hias dari keranjang belanja, jika ya maka program akan memproses untuk menghapus ikan hias yang terpilih dari keranjang, jika tidak maka program dilanjutkan.
- d. *checkout* : User diperintahkan untuk mengkonfirmasi checkout, jika tidak program akan kembali ke menu user. Jika ya maka user diperintahkan untuk menginput uang yang dimiliki. Setelah itu diproses, jika uang lebih dari total harga maka “berhasil checkout”, jika tidak maka program akan mengeluarkan output “uang tidak cukup”.
- e. Logout : admin dapat kembali ke menu utama.

2.3 Konsep/Materi Praktikum yang dipakai

Konsep yang kami gunakan dalam pembuatan laporan ini didasarkan pada materi-materi yang telah dipelajari dalam modul praktikum, di antaranya:

1. **Algoritma dan *Flowchart*** (Modul 1 dan 2)

Sebelum membuat kode program, kami terlebih dahulu menyusun algoritma dan *flowchart* sebagai panduan. Algoritma membantu memastikan setiap langkah logis dalam program terstruktur dengan baik, sedangkan *flowchart* memvisualisasikan alur proses untuk mempermudah pemahaman.

2. **Pengenalan dan Struktur Bahasa Pemrograman Python** (Modul 3 dan 4)

Dalam laporan ini, kami memanfaatkan struktur dasar Python seperti variabel, tipe data, dan operasi dasar untuk membangun program.

Pemahaman mengenai struktur kode, seperti indentasi, sangat penting untuk memastikan program berjalan dengan benar.

3. **Percabangan** (Modul 5)

Percabangan (**if, elif, else**) digunakan untuk membuat keputusan berdasarkan kondisi tertentu. Konsep ini memungkinkan program untuk mengeksekusi blok kode yang berbeda sesuai dengan input atau situasi.

4. **Perulangan** (Modul 6)

Perulangan (**for, while**) digunakan untuk mengulang bagian kode tertentu hingga syarat tertentu terpenuhi. Penggunaan perulangan sangat membantu dalam mengolah data yang berulang atau jumlah yang banyak.

5. **List dan Tuple** (Modul 7)

Kami memanfaatkan struktur data seperti list dan tuple untuk menyimpan data yang bersifat kolektif. List memungkinkan data diubah, sedangkan tuple digunakan untuk data yang bersifat tetap.

6. **Dictionary** (Modul 8)

Dictionary digunakan untuk menyimpan pasangan *key-value*, yang mempermudah pengaksesan data berdasarkan kunci unik.

7. **Fungsi dan Prosedur** (Modul 9)

Fungsi digunakan untuk memisahkan logika program menjadi bagian-bagian yang lebih kecil, modular, dan mudah dikelola. Kami menggunakan fungsi untuk mengurangi redundansi kode dan mempermudah *debugging*.

8. **Error Handling dan File Eksternal** (Modul 10)

Dalam laporan ini, kami juga menerapkan konsep penanganan error menggunakan **try**, **except**, dan **finally**. Selain itu, manipulasi file eksternal seperti membaca dan menulis file dilakukan untuk menyimpan data program.

9. **Git dan GitHub** (Modul 11)

Untuk pengumpulan tugas posttest, kami memanfaatkan Git dan GitHub. Penggunaan platform ini memastikan bahwa perubahan kode

terdokumentasi dengan baik dan memungkinkan kolaborasi yang lebih efektif.

BAB III

HASIL DAN PEMBAHASAN

3.1 Tampilan Program

1. Tampilan Menu Utama



Gambar 3. 1

Program dimulai dengan menu utama tersebut dimana pengguna bisa memilih untuk *login* atau *register* sebuah akun atau keluar dari program. Pengguna bisa menggunakan tombol *Arrow Keys*, lebih tepatnya tombol atas dan bawah untuk navigasi menu, dan tombol *Enter Key* untuk memilih sebuah pilihan menu.

2. Tampilan Login

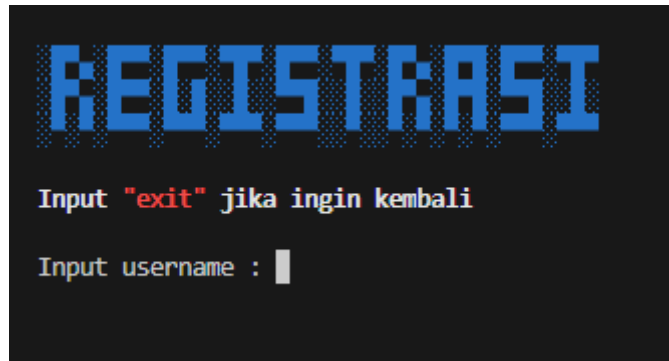


Gambar 3. 2

Di sini, pengguna diberitahukan untuk menginput *username* dan *password* akunnya. Jika akun tersedia, maka pengguna akan dialihkan ke menu *user* atau menu *admin* tergantung *role* akun yang dia masukkan. Jika

akun tidak tersedia atau salah satu kredensial *login* salah, maka akses akan ditolak dan pengguna dipersilahkan untuk menginput *username* dan *password* lagi. Jika pengguna ingin keluar dari menu ini, maka dia bisa menginput “*exit*” ke *input* agar bisa dialihkan ke menu utama.

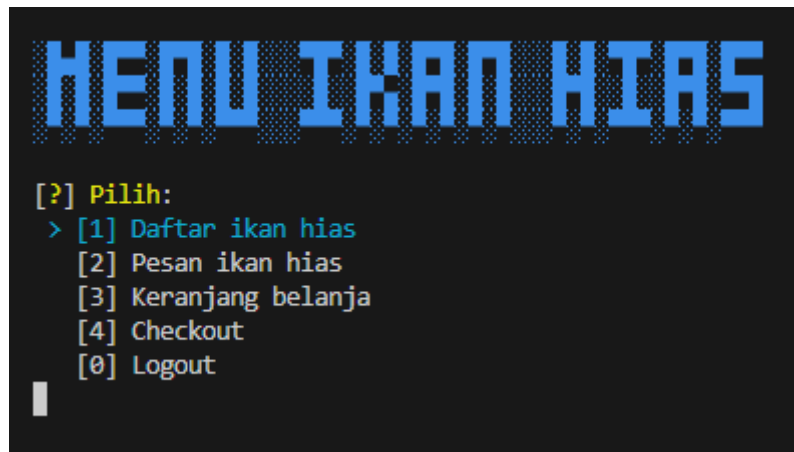
3. Tampilan *Register*



Gambar 3. 3

Di sini, pengguna akan diberitahukan untuk menginput sebuah *username* dan *password*. Jika *username* yang diinput tidak ada di data akun, maka akun akan dibuat dengan *username* dan *password* yang telah diinput pengguna. Jika *username* yang diinput ada di data akun, maka pembuatan akun baru akan ditolak dan pengguna dipersilahkan untuk mencoba membuat akun baru lagi. Seperti pada tampilan *login*, jika pengguna ingin keluar dari menu ini, maka dia bisa menginput “*exit*” ke *input* agar bisa dialihkan ke menu utama.

4. Tampilan Menu *User*



Gambar 3. 4

Jika *role* akun pengguna yang dimasukkan pada saat menu *login* adalah *user*, maka pengguna akan dialihkan ke menu ini. Di sini, pengguna bisa melihat daftar ikan hias, memesan ikan hias, melihat keranjang belanjaan pengguna, dan melakukan pembayaran dengan “*Checkout*”. Pengguna juga bisa beralih ke menu utama dengan pilihan “*Logout*”.

5. Tampilan Daftar Ikan Hias

No	Jenis Ikan	Kelangkaan	Harga	Stok
1	Ikan Cupang	Umum	50000	15
2	Ikan Koi	Agak Langka	75000	8
3	Ikan Mas Koki	Umum	40000	25
4	Ikan Guppy	Umum	20000	12
5	Ikan Arwana	Sangat Langka	300000	4
6	Ikan Discus	Langka	85000	12
7	Ikan Zebra Danio	Umum	30000	30
8	Ikan Pleco	Umum	20	20

Tekan enter untuk lanjut...

Gambar 3. 5

Di sini, pengguna bisa melihat ikan-ikan hias apa saja yang tersedia dijual lengkap dengan kelangkaan, harga, dan stok ikan. Pengguna bisa menekan tombol “Enter Key” untuk kembali ke menu *user*.

6. Tampilan Pesan Ikan Hias

No	Jenis Ikan	Kelangkaan	Harga	Stok
1	Ikan Cupang	Umum	50000	15
2	Ikan Koi	Agak Langka	75000	8
3	Ikan Mas Koki	Umum	40000	25
4	Ikan Guppy	Umum	20000	12
5	Ikan Arwana	Sangat Langka	300000	4
6	Ikan Discus	Langka	85000	12
7	Ikan Zebra Danio	Umum	30000	30
8	Ikan Pleco	Umum	20	20

Input "exit" jika ingin kembali

Pilih nomor ikan hias yang ingin dipesan: 2

Jumlah ikan hias yang ingin dipesan:

Gambar 3. 6

Di sini, pengguna bisa memesan sebuah ikan beserta jumlah ikan yang bisa dibeli. Jika jumlah ikan hias melebihi stok, maka pemesanan akan ditolak dan pengguna dipersilahkan untuk menginput lagi. Pengguna bisa kembali ke menu *user* tanpa memesan apapun dengan menginput "Exit".

7. Tampilan Keranjang Belanja

No	Jenis Ikan	Kelangkaan	Harga	Jumlah
1	Ikan Zebra Danio	Umum	30000	2
2	Ikan Pleco	Umum	25000	10

Input "exit" jika ingin kembali

Ingin menghapus ikan hias dari keranjang? (y/t) :

Gambar 3. 7

Di sini, pengguna bisa melihat keranjang pembelian akun yang berisi apa yang pengguna telah pesan. Setiap akun *user* memiliki keranjangnya sendiri dan tidak pernah akan melihat keranjang pembelian akun *user* lain. Pengguna juga bisa menghapus ikan dari keranjang jika tidak jadi ingin membeli ikan hias tersebut. Pengguna bisa keluar dari tampilan ini tanpa menghapus ikan apapun dengan menginput "exit" atau "y" ke dalam konfirmasi penghapusan ikan.

8. Tampilan Checkout

PEMBAYARAN

Input "exit" jika ingin kembali

Total Harga : Rp55000

Yakin ingin melakukan pembayaran (y/t) ?

Gambar 3. 8

Di sini, pengguna bisa melakukan pembayaran dengan menginput "y" ke konfirmasi pembayaran dan menginput uang yang akan dibayar. Jika input uang kurang dari total harga, maka pembayaran akan ditolak. Jika input uang lebih dari total harga, maka program akan melaksanakan akan menunjukkan uang angsulan. Jika input uang sama dengan total harga, maka program hanya akan melaksanakan pembayaran tanpa menunjukkan uang

angsuran. Pengguna bisa keluar dari tampilan ini tanpa membayar apapun dengan menginput “exit” atau “y” ke dalam konfirmasi pembayaran.

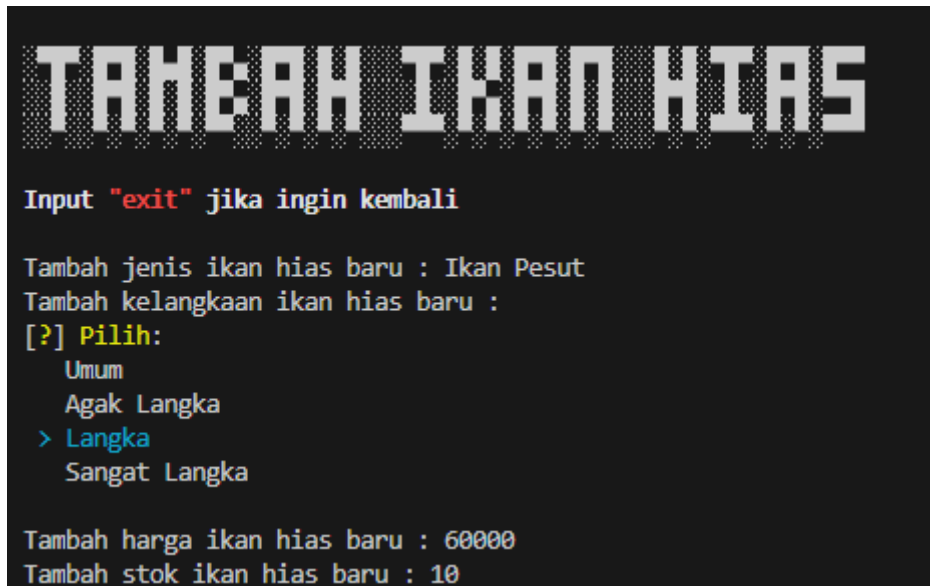
9. Tampilan Menu Admin



Gambar 3. 9

Jika *role* akun pengguna yang dimasukkan pada saat menu *login* adalah ADMIN, maka pengguna akan dialihkan ke menu ini. Di sini, pengguna bisa melakukan CRUD (*Create, Read, Update, Delete*) pada data ikan. Seperti Menu *user*, Pengguna juga bisa beralih ke menu utama dengan pilihan “Logout”.

10. Tampilan Penambahan Ikan Hias



```
TAMBAH IKAN HIAS

Input "exit" jika ingin kembali

Tambah jenis ikan hias baru : Ikan Pesut
Tambah kelangkaan ikan hias baru :
[?] Pilih:
    Umum
    Agak Langka
    > Langka
    Sangat Langka

Tambah harga ikan hias baru : 60000
Tambah stok ikan hias baru : 10
```

Gambar 3. 10

Di sini, pengguna bisa menambahkan jenis ikan baru ke data dengan menginput jenis ikan, kelangkaan ikan, harga ikan, dan stok ikan baru. Jika pengguna tidak ingin menambahkan ikan, maka pengguna bisa menginput “exit” untuk kembali ke menu admin

11. Tampilan Pembacaan Ikan Hias

No	Jenis Ikan	Kelangkaan	Harga	Stok
1	Ikan Cupang	Umum	50000	20
2	Ikan Koi	Agak Langka	75000	9
3	Ikan Mas Koki	Umum	40000	25
4	Ikan Guppy	Umum	20000	22
5	Ikan Neon Tetra	Umum	15000	20
6	Ikan Arwana	Sangat Langka	300000	5
7	Ikan Discus	Langka	85000	12
8	Ikan Louhan	Langka	120000	16
9	Ikan Zebra Danio	Umum	30000	29
10	Ikan Pleco	Umum	25000	15
11	Ikan Pesut	Langka	60000	10

Gambar 3. 11

Seperti pada tampilan daftar ikan hias di menu *user*, Di sini, pengguna bisa melihat ikan-ikan hias apa saja yang tersedia dijual lengkap dengan kelangkaan, harga, dan stok ikan. Pengguna bisa menekan tombol “Enter Key” untuk kembali ke menu *user*.

12. Tampilan Pembaruan Ikan Hias

No	Jenis Ikan	Kelangkaan	Harga	Stok
1	Ikan Cupang	Umum	50000	20
2	Ikan Koi	Agak Langka	75000	9
3	Ikan Mas Koki	Umum	40000	25
4	Ikan Guppy	Umum	20000	22
5	Ikan Neon Tetra	Umum	15000	20
6	Ikan Arwana	Sangat Langka	300000	5
7	Ikan Discus	Langka	85000	12
8	Ikan Louhan	Langka	120000	16
9	Ikan Zebra Danio	Umum	30000	29
10	Ikan Pleco	Umum	25000	15
11	Ikan Pesut	Sangat Langka	60000	10

Input "exit" jika ingin kembali

Pilih nomor ikan hias yang ingin diupdate : 11

Gambar 3. 12

Di tampilan menu ini, pengguna dipersilahkan untuk mengupdate jenis ikan dengan data baru dengan memilih nomor ikan. Jika tidak ingin mengubah salah satu data ikan, pengguna bisa membiarkan *input* kosong agar tidak diubah. Pengguna bisa keluar tanpa mengupdate data apapun dengan menginput "exit".

13. Tampilan Penghapusan Ikan Hias

No	Jenis Ikan	Kelangkaan	Harga	Stok
1	Ikan Cupang	Umum	50000	20
2	Ikan Koi	Agak Langka	75000	9
3	Ikan Mas Koki	Umum	40000	25
4	Ikan Guppy	Umum	20000	22
5	Ikan Neon Tetra	Umum	15000	20
6	Ikan Arwana	Sangat Langka	300000	5
7	Ikan Discus	Langka	85000	12
8	Ikan Louhan	Langka	120000	16
9	Ikan Zebra Danio	Umum	30000	29
10	Ikan Pleco	Umum	25000	15

Input "exit" jika ingin kembali

Pilih nomor ikan hias yang ingin dihapus :

Gambar 3. 13

Di tampilan menu ini, pengguna bisa menghapus salah satu jenis ikan hias di dalam data dengan menginput nomor ikan. Jika ingin keluar dari tampilan menu ini tanpa menghapus ikan apapun dari data, pengguna bisa menginput "exit" untuk keluar ke menu admin.

3.2 Source Code

```
import os #menyediakan puluhan fungsi untuk berinteraksi
dengan sistem operasi
from tabulate import tabulate # library untuk tampilan
tabel
import random # library untuk mendapatkan nilai random
import time #library untuk jeda time sleep
from tqdm import tqdm # library untuk loading bar
```

```

from typing import Callable # library untuk type hints
import inquirer # library untuk menu interaktif
import csv # library untuk mengelola data eksternal csv
from pathlib import Path

# Path file relatif terhadap lokasi script
daftar_ikan_path = Path(__file__).parent /
'daftar_ikan.csv'
akun_path = Path(__file__).parent / 'akun.csv'
keranjang_path = Path(__file__).parent / 'keranjang.csv'

# Warna
BOLD = '\033[1m'
RED = '\033[31m'
GREEN = '\033[32m'
YELLOW = '\033[33m'
BLUE = '\033[34m'
WHITE = '\033[97m'
RESET = '\033[0m'

# clear terminal
def clear():
    os.system("cls || clear")

# validasi input bertipe string
def validasi_input_huruf(charInput:str) -> bool:
    for char in charInput:
        # cek apakah karakter bukan karakter atau spasi
        if not (char.isalpha() or char.isspace()):
            return False
    return True

# validasi input bertipe number
def validasi_input_angka(numInput:int) -> bool:
    # True jika numInput sepenuhnya angka
    return numInput.isdigit()

# fake loading
def loading(loadingLength:int, delay:int):

```

```

        for i in tqdm(range(loadingLength)):
            time.sleep(delay)

# fake loading (2)
# ambil nilai random dari 1-2
random = random.randint(1, 2)
def wait(index:int):
    clear()
    print("Loading", end='')
    # perulangan untuk menampilkan efek loading berupa
    titik-titik sampai 3
    for i in range(3):
        # buat agar titik langsung ditampilkan setelah
        jeda
        print(".", end='', flush=True)
        time.sleep(0.2)

    # selama index lebih kecil dari nilai random maka akan
    melakukan rekursif
    if index < random:
        wait(index+1)

def pesanInputExit():
    print(f'{BOLD}Input {RED}"exit"{WHITE} jika ingin
    kembali{RESET}\n')

# menahan dan melanjutkan program
def lanjut():
    print(BOLD+"Tekan enter untuk lanjut..." +RESET)
    input()
    clear()

# fungsi untuk validasi input
def validasiInput(prompt:str, fungsiValidasi:
Callable[[str], bool], pesanError:str,
cekJenisValidasi:str, *valueNoUpdate):
    while True:
        try:
            # menampilkan pesan input

```

```

        inputVariable = input(prompt).strip()
        # cek apakah isi input sama dengan 'exit'
        if inputVariable == 'exit':
            # cek apakah 'admin' terdapat di dalam
cekJenisValidasi
            if 'admin' in cekJenisValidasi:
                menuAdmin()
            else:
                menuUser()
        else:
            # cek apakah 'update' terdapat di dalam
cekJenisValidasi
            # setiap cekJenisValidasi yang memiliki
nilai argumen 'update' berarti jenis inputnya merupakan
untuk melakukan update
            if 'update' in cekJenisValidasi :
                # jika input kosong
                if not inputVariable:
                    # buat agar input kosong tetap
diisi oleh data saat ini
                    inputVariable =
str(*valueNoUpdate)
                else:
                    if not inputVariable:
                        raise ValueError("Tidak boleh
kosong")

            # cek apakah 'angka' terdapat di dalam
pesanError
            if 'angka' in pesanError:
                # jika input diawali '-' dan diikuti
angka
                if inputVariable.startswith("-") and
inputVariable[1:].isdigit():
                    # keluarkan error tidak bisa
negatif
                    raise ValueError("Hanya bisa
bilangan positif")

```

```

        if not fungsiValidasi(inputVariable) or '-'
' in inputVariable:
            raise ValueError(pesanError)

        return inputVariable

    except ValueError as error:
        print(f"{RED}{BOLD}Input tidak valid: {error}
Silakan coba lagi.\n{RESET}")

# fungsi untuk memunculkan opsi pilihan
def opsiMenu(*banyakOpsi):
    global answers
    totalOpsi = []
    # perulangan mengambil tiap opsi lalu menaruhnya di
dalam list totalOpsi
    for opsi in banyakOpsi:
        totalOpsi.append(opsi)
    option = [
        inquirer.List(
            "opsi",
            message=YELLOW+"Pilih"+RESET,
            choices=[
                # melakukan unpacking
                *totalOpsi
            ],
        ),
    ]

    # mendapatkan jawaban
    answers = inquirer.prompt(option)

def tampilkanIkan():
    try:
        with open(daftar_ikan_path, "r") as file:
            reader = csv.DictReader(file)

            # Mengonversi seluruh data CSV menjadi sebuah
dictionary

```



```

        # Gunakan nama ikan (Jenis Ikan) sebagai key,
        dan data lainnya sebagai nilai (value) dalam dictionary
        # membaca tiap baris berdasarkan nama
        headernya

        dataIkanHias = {row["Jenis Ikan"]:
{"Kelangkaan": row["Kelangkaan"], "Harga": row["Harga"],
"Stok": row["Stok"]}}

        for row in reader}

    if dataIkanHias:
        headers = ["No", "Jenis Ikan",
"Kelangkaan", "Harga", "Stok"]

        # Buat table dengan dictionary
        table = [
            # jenis_ikan sebagai key, dan data
            adalah valuenya
            [index + 1, jenis_ikan,
data["Kelangkaan"], data["Harga"], data["Stok"]]
            for index, (jenis_ikan, data) in
            enumerate(dataIkanHias.items())
        ]

        # Menampilkan tabel menggunakan tabulate
        print(tabulate(table, headers,
tablefmt="grid"))
    else:
        print("Data kosong.")

except FileNotFoundError:
    print("File tidak ditemukan, buat data baru
terlebih dahulu.")

# ----- PROGRAM UNTUK MENU ADMIN -----
-----

# admin menambah data ikan hias baru
def admin_tambahIkan():
    print('')

```

```

'''

    pesanInputExit()
    # input tambah jenis ikan
    jenisIkan = validasiInput("Tambah jenis ikan hias baru
: ", validasi_input_huruf, "Hanya bisa huruf", 'admin
create')

    # input opsi tambah kelangkaan ikan
    print("Tambah kelangkaan ikan hias baru : ")
    opsiMenu('Umum', 'Agak Langka', 'Langka', 'Sangat
Langka')
    kelangkaan = answers['opsi']

    # input tambah harga ikan
    hargaIkan = validasiInput("Tambah harga ikan hias baru
: ", validasi_input_angka, "Hanya bisa angka", 'admin
create')

    # input tambah stok ikan
    stok = validasiInput("Tambah stok ikan hias baru : ",
validasi_input_angka, "Hanya bisa angka", 'admin create')

    # membuka file daftar_ikan.csv, dan menambah data baru
dengan metode 'a'/append
    with open(daftar_ikan_path, 'a', newline='') as file:
        writer = csv.writer(file)
        writer.writerow([jenisIkan, kelangkaan, hargaIkan,
stok])

    with open(daftar_ikan_path, 'r') as file:
        reader = csv.reader(file)
        dataIkanHias = list(reader)

```

```

delay = 0.1
if len(dataIkanHias) > 5:
    delay = 0.07
elif len(dataIkanHias) > 10:
    delay = 0.05
elif len(dataIkanHias) > 15:
    delay = 0.03
elif len(dataIkanHias) > 20:
    delay = 0.01

loading(len(dataIkanHias), delay)
print(GREEN+BOLD+"\nBerhasil menambah data ikan
hias!"+RESET)
lanjut()

# admin mengupdate data ikan hias
def admin_updateIkan():
    tampilkanIkan()
    pesanInputExit()
    while True:
        try:
            # membaca file daftar_ikan.csv
            with open(daftar_ikan_path, "r") as file:
                # mengubah data csv menjadi list
                dataIkanHias = list(csv.reader(file))

            index = validasiInput(YELLOW+BOLD+"Pilih nomor
ikan hias yang ingin diupdate : "+RESET,
validasi_input_angka, "Hanya bisa angka", 'admin index')

            # jika berhasil melewati error, ubah value
index menjadi integer
            index = int(index)

            # Pastikan index berada dalam rentang yang
valid
            if 0 < index < len(dataIkanHias):
                # unpacking

```

```

        jenisIkan, kelangkaan, hargaIkan, stok =
dataIkanHias[index]

        # input mengubah kelangkaan ikan
(optional)

        jenisIkan_baru = validasiInput(f"Ubah
jenis ikan hias {BOLD}(enter jika tidak ingin
mengubah){RESET} : ", validasi_input_huruf, "Hanya bisa
huruf", 'admin update', jenisIkan)

        # input opsi untuk mengubah kelangkaan
ikan (wajib memilih)

        print("Ubah kelangkaan ikan hias : ")
        opsiMenu('Umum', 'Agak Langka', 'Langka',
'Sangat Langka')
        kelangkaan_baru = answers['opsi']

        # input mengubah harga ikan (opsional)
        hargaIkan_baru = validasiInput(f"Ubah
harga ikan hias {BOLD}(enter jika tidak ingin
mengubah){RESET} : ", validasi_input_angka, "Hanya bisa
angka", 'admin update', hargaIkan)

        # input mengubah stok ikan (opsional)
        stok_baru = validasiInput(f"Ubah stok ikan
hias {BOLD}(enter jika tidak ingin mengubah){RESET} : ",
validasi_input_angka, "Hanya bisa angka", 'admin update',
stok)

        # mengambil data yang dipilih
        dataIkanHias[index] = jenisIkan_baru,
kelangkaan_baru, hargaIkan_baru, stok_baru

        # mengubah isi dari dataIkanHias yang
dipilih

        with open(daftar_ikan_path, "w",
newline='') as file:
            writer = csv.writer(file)
            writer.writerows(dataIkanHias)

```

```

        loading(10, 0.05)
        print(GREEN+BOLD+"\nBerhasil update data
ikan hias!" + RESET)

        lanjut()
        break

    else:
        raise ValueError("Ikan hias tidak
ditemukan!")

except ValueError as error:
    print(f"{RED}{BOLD}Input tidak valid: {error}
Silakan coba lagi.\n{RESET}")

    # bila file tidak ditemukan maka muncul pesan
error dengan menangkap error FileNotFoundError
    except FileNotFoundError:
        print("File tidak ditemukan, buat data baru
terlebih dahulu.")

# admin menghapus data ikan hias
def admin_hapusIkan():
    tampilkanIkan()
    pesanInputExit()
    # membaca file daftar_ikan.csv
    with open(daftar_ikan_path, "r") as file:
        # mengubah data csv menjadi list
        dataIkanHias = list(csv.reader(file))

    while True:
        try:
            index = validasiInput(YELLOW+BOLD+"Pilih nomor
ikan hias yang ingin dihapus : " + RESET,
validasi_input_angka, "Hanya bisa angka", 'admin index')
            index = int(index)

            # Pastikan index berada dalam rentang yang
valid

            if 0 <= index - 1 < len(dataIkanHias):

```

```

        # Hapus data dari dataIkanHias
        del dataIkanHias[index]

        # Tulis ulang dataIkanHias yang sudah di-
update ke daftar_ikan.csv
        with open(daftar_ikan_path, "w",
newline='') as file:

csv.writer(file).writerows(dataIkanHias)

        loading(10, 0.05)
        print(GREEN+BOLD+"\nIkan hias berhasil
dihapus"+RESET)
        lanjut()
        break

    except ValueError as error:
        print(f"{RED}{BOLD}Input tidak valid: {error}
Silakan coba lagi.\n{RESET}")

# ----- PROGRAM UNTUK MENU USER -----
-----

# user memesan ikan hias
def user_pesanIkanHias():
    tampilkanIkan()
    pesanInputExit()
    # membaca file daftar_ikan.csv
    with open(daftar_ikan_path, "r") as file:
        # mengubah data csv menjadi list
        dataIkanHias = list(csv.reader(file))

    while True:
        try:
            pilihan = validasiInput(YELLOW+BOLD+"Pilih
nomor ikan hias yang ingin dipesan: "+RESET,
validasi_input_angka, "Hanya bisa angka", 'user index')
            pilihan = int(pilihan)

```

```

        # Pastikan pilihan berada dalam rentang yang
valid
        if 1 <= pilihan <= len(dataIkanHias)-1:
            jmlhpesanan =
validasiInput(YELLOW+BOLD+"Jumlah ikan hias yang ingin
dipesan: "+RESET, validasi_input_angka, "Hanya bisa
angka", 'user jumlah')
            jmlhpesanan = int(jmlhpesanan)

            # digunakan agar keaslian data pada
daftar_ikan.csv tidak berubah sehingga mudah ketika
menulis ulang data menggunakan 'w'
            ikanTerpilih = dataIkanHias[pilihan]
            # buat copy list agar nanti digunakan
hanya untuk dimasukkan ke data keranjang.csv
            ikanTerpilih_copy =
dataIkanHias[pilihan].copy()

            # Membaca keranjang untuk mencari apakah
ikan sudah ada di dalamnya
            keranjang_data = []
            with open(keranjang_path, 'r') as file:
                keranjang_data =
list(csv.reader(file))
            # Filter data berdasarkan username
yang saat ini sedang login
            keranjang_data_login = [baris for
baris in keranjang_data if baris[-1] ==
username_sedangLogin]

            # jumlah pesanan harus lebih kecil atau
sama dengan stok
            if jmlhpesanan <= int(ikanTerpilih[3]):
                # Untuk mengecek apakah ikan sudah ada
di keranjang
                ikan_ditemukan = False
                for data in keranjang_data_login:
                    if data and data[0] ==
ikanTerpilih_copy[0]: # Cek jenis ikan

```

```

        data[3] = int(data[3]) +
jmlhpesanan # Tambah jumlah pesanan
        ikan_ditemukan = True
        break

        # Jika ikan belum ada di keranjang,
tambahkan data baru
        if not ikan_ditemukan:
            # insert jumlah pesanan ke kolom
ke 4
            ikanTerpilih_copy.insert(3,
jmlhpesanan)
            # insert username yang sedang
memesan ke kolom ke 5
            ikanTerpilih_copy.insert(4,
username_sedangLogin)
            # tambahkan 5 kolom data

keranjang_data.append(ikanTerpilih_copy[:5])

        # Tulis kembali ke keranjang.csv
        with open(keranjang_path, 'w',
newline='') as file:
            writer = csv.writer(file)
            writer.writerows(keranjang_data)

            ikanTerpilih[3] =
int(ikanTerpilih[3])
            ikanTerpilih[3] -= jmlhpesanan

        loading(10, 0.05)

print(f"\n{GREEN}{BOLD}'{dataIkanHias[pilihan][0]}' telah
ditambahkan ke keranjang.{RESET}")

        if ikanTerpilih[3] < 1:
            # hapus ikan hias yang dipilih
            del dataIkanHias[pilihan]

```



```

        # Tulis ulang dataIkanHias yang sudah
        di-update ke daftar_ikan.csv
        with open(daftar_ikan_path, 'w',
newline='') as file:

csv.writer(file).writerows(dataIkanHias)

        lanjut()
        break

    else:
        print(RED+BOLD+"Jumlah pesanan
melebihi ketersediaan stok ikan.\n"+RESET)

    else:
        print(RED+BOLD+"Ikan hias yang dipilih
tidak ada.\n"+RESET)

except ValueError as error:
    print(f"{RED}{BOLD}Input tidak valid: {error}
Silakan coba lagi.\n{RESET}")

# user tampilkan keranjang belanja
def user_keranjangBelanjaan():
    try:
        # Membaca file CSV dan memproses data
        with open(keranjang_path, "r") as file:
            readerKeranjang = csv.reader(file)

            # Mengubah data CSV menjadi sebuah list
            dataIkanHias = list(readerKeranjang)

            # Filter data berdasarkan username_sedangLogin
            dataIkanHias_login = [baris for baris in
dataIkanHias if baris[-1] == username_sedangLogin]
            dataIkanHias_tidakLogin = [baris for baris in
dataIkanHias if baris[-1] != username_sedangLogin]

            if len(dataIkanHias_login) == 0:

```

```

        print('Keranjang belanja kosong, silahkan
pesan terlebih dahulu 😊')

    else:
        tableRow = []
        for i, data in
enumerate(dataIkanHias_login):
            row = i+1,*data[:4]
            tableRow.append(row)

        # Menampilkan hasil akhir dalam bentuk
tabel

        headers = ["No", "Jenis Ikan",
"Kelangkaan", "Harga", "Jumlah"]
        print(tabulate(tableRow, headers,
tablefmt="grid"))

    pesanInputExit()

    while True:
        # konfirmasi apakah ingin menghapus
ikan dari keranjang belanja
        konfirInginHapus =
input(YELLOW+BOLD+"Ingin menghapus ikan hias dari
keranjang? (y/t) : "+RESET).strip().lower()

        if konfirInginHapus in ["y", "t"]:
            break
        elif konfirInginHapus == 'exit':
            menuUser()
        else:
            print("Input tidak valid, hanya
bisa (y/t).")

    while True:
        # Hapus ikan dari keranjang
        if konfirInginHapus == "y":

```

```

        hapusDariKeranjang =
validasiInput("No ikan hias yang ingin dihapus : ",
validasi_input_angka, "Hanya bisa angka", 'user index')
        hapusDariKeranjang =
int(hapusDariKeranjang)-1

        # cek apakah nomor yang dipilih
ada
        if 1 <= hapusDariKeranjang <=
len(tableRow):
            with open(daftar_ikan_path,
"r") as file:
                dataIkanHias_InDaftar =
list(csv.reader(file))

                for data in
dataIkanHias_InDaftar:
                    # cek jenis ikan
apakah sama dengan jenis ikan yang dipilih
                    if data[0] ==
dataIkanHias_login[hapusDariKeranjang][0]:
                        # ubah kembali
data stok pada daftar_ikan.csv
                        # dengan
menambah stok saat ini dan stok pada ikan hias yang
dipilih untuk dihapus dari keranjang

                        # semisal stok
saat ini 3 dan stok pada ikan hias yang dipilih 2 maka
stok pada
                        #
daftar_ikan.csv akan menambah lagi
                        data[3] =
int(data[3]) +
int(dataIkanHias_login[hapusDariKeranjang][3])

                    # Tulis ulang
dataIkanHias_InDaftar yang sudah di-update ke
daftar_ikan.csv

```

```

        with open(daftar_ikan_path,
"w", newline='') as file:

            writer = csv.writer(file)

writer.writerow(dataIkanHias_InDaftar)

            # Hapus ikan hias yang dipilih
dari keranjang.csv

            del
dataIkanHias_login[hapusDariKeranjang]

            # Tulis ulang dataIkanHias ke
keranjang.csv

            with open(keranjang_path, "w",
newline='') as file:

                writer = csv.writer(file)

writer.writerow(dataIkanHias_login)

writer.writerow(dataIkanHias_tidakLogin)

            print(GREEN+BOLD+"\nBerhasil
menghapus ikan hias dari keranjang"+RESET)
            break
        else:
            print("Ikan hias tidak
ditemukan!")
    else:
        break

    # bila file tidak ditemukan maka muncul pesan error
dengan menangkap error FileNotFoundError
    except FileNotFoundError:
        print("File tidak ditemukan, buat data baru
terlebih dahulu.")
        lanjut()

# user melakukan pembayaran
def user_checkout():

```

```

print('''
████████████████████████████████████████████████████████████████████████████████
████████████████████████████████████████████████████████████████████████████████
████████████████████████████████████████████████████████████████████████████████
''')

pesanInputExit()

try:
    with open(keranjang_path, "r") as file:
        reader = csv.reader(file)

        # Mengubah data CSV menjadi sebuah list
        dataIkanHias = list(reader)

        # Filter data berdasarkan username yang saat
        ini sedang login
        dataIkanHias_login = [baris for baris in
dataIkanHias if baris[-1] == username_sedangLogin]

        # Filter data berdasarkan username yang saat
        ini tidak login
        dataIkanHias_tidakLogin = [baris for baris in
dataIkanHias if baris[-1] != username_sedangLogin]

        # Mengambil semua nilai dari baris ke-3
        (indeks 2) yaitu harga
        kolom_harga = [int(baris[2]) for baris in
dataIkanHias if baris[-1] == username_sedangLogin]

        # Menghitung jumlah total dari kolom harga
        jumlah = sum(kolom_harga)

        print(f"Total Harga : Rp{jumlah}")

        if jumlah > 0:
            while True:
                konfirPembayaran =
input(YELLOW+BOLD+"Yakin ingin melakukan pembayaran (y/t)
? "+RESET).strip().lower()

```

```

        if konfirPembayaran == "y":
            bayar = validasiInput("Input uang
yang dimiliki : ", validasi_input_angka, "Hanya bisa
angka", 'nominal')

            bayar = int(bayar)

            loading(10, 0.05)

            if bayar >= jumlah:
                print(GREEN+BOLD+"\nBerhasil
melakukan pembayaran"+RESET)
                # hapus data pada keranjang
ketika berhasil bayar

                del dataIkanHias_login

                # tulis kembali data yang
sudah kosong tadi ke dalam keranjang.csv
                with open(keranjang_path, "w",
newline='') as file:
                    writer = csv.writer(file)
                    for baris in
dataIkanHias_tidakLogin:
                        writer.writerow(baris)

                    kembalian = bayar - jumlah
                    if kembalian > 0:

print(f"\n{GREEN}{BOLD}Uang kembalian : Rp{kembalian},
Terimakasih sudah berbelanja🥰{RESET}")
                        else:

print(GREEN+BOLD+"\nTerimakasih sudah
berbelanja🥰"+RESET)
                            else:
                                print(RED+BOLD+"\nMaaf uang
anda tidak cukup🥰"+RESET)

                                break
                            elif konfirPembayaran == "t":

```



```

        admin_tambahIkan()
    elif '[2]' in pilihan:
        clear()
        tampilkanIkan()
        lanjut()
    elif '[3]' in pilihan:
        clear()
        admin_updateIkan()
    elif '[4]' in pilihan:
        clear()
        admin_hapusIkan()
    elif '[0]' in pilihan:
        clear()
        print("Logout")
        time.sleep(1)
        menuUtama()
except TypeError:
    print(f"{RED}{BOLD}Program terhenti{RESET}")
    menuAdmin()

# menu user
def menuUser():
    clear()
    while True:
        print(BLUE+BOLD+'''
┌───┴───┐ ┌───┴───┐ ┌───┴───┐ ┌───┴───┐ ┌───┴───┐ ┌───┴───┐ ┌───┴───┐
└───┬───┘ └───┬───┘ └───┬───┘ └───┬───┘ └───┬───┘ └───┬───┘ └───┬───┘
      │       │       │       │       │       │       │       │
''' + RESET)
        opsiMenu("[1] Daftar ikan hias", "[2] Pesan ikan
hias", "[3] Keranjang belanja", "[4] Checkout", "[0]
Logout")

    try:
        pilihan = answers["opsi"]

        # cek apakah ada string '[nomor]' di dalam
variabel pilihan, jika ada jalankan sesuai kondisi
        if '[1]' in pilihan:
```



```

        clear()
        tampilkanIkan()
        lanjut()
    elif '[2]' in pilihan:
        clear()
        user_pesanIkanHias()
    elif '[3]' in pilihan:
        clear()
        user_keranjangBelanjaan()
    elif '[4]' in pilihan:
        clear()
        user_checkout()
        clear()
    elif '[0]' in pilihan:
        clear()
        print("Logout")
        time.sleep(1)
        menuUtama()
except TypeError:
    print(f"{RED}{BOLD}Program terhenti{RESET}")
    menuUser()

# ----- PROGRAM LOGIN DAN REGISTRASI -----
-----

# login
def login():
    global username_sedangLogin # Menandai bahwa kita
    menggunakan variabel global
    clear()
    print(BLUE+BOLD+'''
███ ███ ███ ███ ███
███ ███ ███ ███ ███
███ ███ ███ ███ ███
''' + RESET)
    pesanInputExit()
    while True:
        try:
            # error handling pada input

```

```

        inputUsername = input("Username anda :
").strip()
        if not inputUsername:
            raise ValueError("Username tidak boleh
kosong")
        elif inputUsername == 'exit':
            menuUtama()

        inputPassword = input("Password anda :
").strip()
        if not inputPassword:
            raise ValueError("Password tidak boleh
kosong")
        elif inputPassword == 'exit':
            menuUtama()

        with open(akun_path, "r") as file:
            reader = csv.reader(file)
            dataAkun = list(reader)

            # Memeriksa setiap akun
            for data in dataAkun[1:]:
                username, password, role = data
                if inputUsername == username and
inputPassword == password:
                    # simpan username yang saat ini
sedang login
                    username_sedangLogin = username
                    if int(role) == 1:
                        wait(0)
                        print(RED+BOLD+"\nLogin
sebagai admin 🏠"+RESET)
                        time.sleep(1)
                        menuAdmin()
                    elif int(role) == 0:
                        wait(0)
                        print(RED+BOLD+"\nLogin
sebagai user 👤"+RESET)
                        time.sleep(1)

```

```

menuUser()

# jika akun tidak ditemukan
raise ValueError("Username atau password
salah!")

except ValueError as error:
    print(f"{RED}{BOLD}Input tidak valid: {error}
Silakan coba lagi.\n{RESET}")

# registrasi akun user baru
def registrasi():
    clear()
    print(BLUE+
'''
████████████████████████████████████████████████████████████████████████████████
████████████████████████████████████████████████████████████████████████████████
████████████████████████████████████████████████████████████████████████████████
'''
+RESET)
    pesanInputExit()
    while True:
        try:
            # error handling pada input
            registUsername = input("Input username :
").strip()
            if not registUsername:
                raise ValueError("Username tidak boleh
kosong")
            if registUsername.lower() == 'username':
                raise ValueError("Tidak dapat regist
menggunakan username itu")
            elif registUsername == 'exit':
                menuUtama()

            registPassword = input("Input password :
").strip()
            if not registPassword:
                raise ValueError("Password tidak boleh
kosong")

```

```

        if registPassword.lower() == 'password':
            raise ValueError("Tidak dapat regist
menggunakan password itu")
        elif registPassword == 'exit':
            menuUtama()

# Membaca data dari file CSV
with open(akun_path, "r") as file:
    reader = csv.reader(file)
    dataAkun = list(reader)

# Memeriksa apakah username sudah ada
username_sudahAda = False
for akun in dataAkun:
    username, *data = akun
    if registUsername == username:
        username_sudahAda = True
        break

if username_sudahAda:
    raise ValueError("Username sudah ada!")

# Menambahkan akun baru ke file CSV
with open(akun_path, "a", newline='') as file:
    writer = csv.writer(file)
    writer.writerow([registUsername,
registPassword, 0])

# Efek loading dan pesan berhasil
loading(10, 0.05)
print(GREEN + BOLD + "\nBerhasil melakukan
registrasi" + RESET)
lanjut()
break

except ValueError as error:
    print(f"{RED}{BOLD}Input tidak valid: {error}
Silakan coba lagi.\n{RESET}")

```


BAB IV

PENUTUP

4.1 Kesimpulan

Berdasarkan laporan di atas, berikut adalah beberapa kesimpulan yang dapat diambil:

1. Efisiensi Pengelolaan Data

Sistem yang dirancang memudahkan admin dalam mengelola data ikan hias, seperti menambah, memperbarui, atau menghapus data dengan mudah. Penggunaan file CSV sebagai media penyimpanan juga menjamin persistensi data meskipun program tidak berjalan.

2. Kemudahan Penggunaan oleh User

Antarmuka berbasis teks yang sederhana dan user-friendly memberikan kemudahan bagi user untuk memesan ikan hias, melihat keranjang belanja, dan melakukan transaksi. Fitur checkout memastikan proses transaksi berjalan dengan baik dan stok diperbarui secara otomatis.

3. Penerapan Konsep Pemrograman Python

Proyek ini berhasil mengaplikasikan konsep-konsep pemrograman seperti algoritma, flowchart, percabangan, perulangan, fungsi, manipulasi file, dan struktur data Python. Konsep-konsep ini mendukung pengembangan sistem yang modular, terstruktur, dan mudah dikelola.

4. Penggunaan Git dan GitHub

Dengan memanfaatkan Git dan GitHub, pengelolaan versi kode dan kolaborasi selama pengembangan proyek menjadi lebih efektif, memastikan semua perubahan terdokumentasi dengan baik.

4.2 Saran

Untuk pengembangan lebih lanjut, berikut adalah beberapa saran yang dapat dipertimbangkan:

1. **Pencatatan Riwayat Transaksi**

Menambahkan fitur untuk mencatat dan menampilkan riwayat transaksi, baik untuk user maupun admin, akan memberikan nilai tambah dalam pengelolaan sistem dan transparansi data.

2. **Pengembangan Antarmuka Grafis**

Mengembangkan antarmuka berbasis grafis (GUI) dapat meningkatkan pengalaman pengguna (user experience) dan membuat sistem lebih mudah digunakan oleh berbagai kalangan.

3. **Optimalisasi Penyimpanan Data**

Pertimbangkan untuk menggunakan database seperti SQLite atau MySQL untuk menyimpan data, karena database lebih fleksibel dan efisien dibandingkan file CSV, terutama jika data semakin bertambah besar.

4. **Fitur Laporan Stok dan Penjualan**

Menambahkan fitur untuk menghasilkan laporan stok dan penjualan secara otomatis akan sangat membantu admin dalam menganalisis data dan membuat keputusan strategis.

DAFTAR PUSTAKA

inquirer 3.4.0. (2024, Agustus 12). Retrieved from pypi:

<https://pypi.org/project/inquirer/>

Python Random Module. (n.d.). Retrieved from w3schools:

https://www.w3schools.com/python/module_random.asp

tabulate 0.9.0. (2022, Oktober 07). Retrieved from pypi:

<https://pypi.org/project/tabulate/>

tqdm. (n.d.). Retrieved from tqdm documentation: <https://tqdm.github.io/>

Kelas Terbuka. (2022). *Belajar Python [Dasar] - 32 - Copy List*. [Video].



YouTube. <https://www.youtube.com/watch?v=mATeKWmB7YM&list=PLZS-MHyEIro7cgStrKAMhgnOT66z2qKz1&index=33>

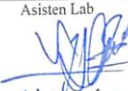

Kelas Terbuka. (2022). *Belajar Python [Dasar] - 49 - Type Hints pada*

Fungsi. [Video].

YouTube. <https://www.youtube.com/watch?v=NR3m8VJA738&list=PLZS-MHyEIro7cgStrKAMhgnOT66z2qKz1&index=51>

LAMPIRAN

Tanggal Konsultasi : 29 Oktober 2024	
<p>Uraian / Pembahasan :</p> <p>Revisi dan saran :</p> <ol style="list-style-type: none"> 1. Konfirmasi pembayaran, ditambahkan input yang ^{untuk} membayar, jika uang yang dibayarkan kurang dari jumlah pembayaran maka pembayaran gagal, jika uang pas atau lebih maka pembayaran berhasil, dengan uang kembalian jika uang lebih. 2. Di bagian keranjang.csu di buat dictionary dgn index key berdasarkan nama user, agar tiap user memiliki data yang masing-masing. 	
Asisten Lab	Ketua Kelompok
	
Nama: Alvin G.S	Nama: Retha Setiawan

Tanggal Konsultasi : 19 November 2024	
<p>Uraian / Pembahasan :</p> <p>Revisi dan Saran :</p> <ol style="list-style-type: none"> 1. Input "x" untuk keluar diganti dengan "exit" atau "keluar" agar lebih jelas dan efektif. 2. time.sleep disarankan hanya menggunakan 0.5 atau 1, lebih dari itu halaman. 3. masing-masing pembagian tugas dikerjakan dan lebih diperdalam. 	
Asisten Lab	Ketua Kelompok
	
Nama: Alvin G.S	Nama: Retha Setiawan

Lampiran 1