

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



MANAJEMEN TOKO HIJAB

Oleh:

Kelompok 6

SYAFIQ HAFIZH FARIZI	2409106009
RUSDIANSYAH	2409106013
NIKY JENITA PUTRI	2409106019

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

KATA PENGANTAR

Puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, kami dapat menyelesaikan proyek akhir yang berjudul "Manajemen Toko Hijab" ini. Proyek ini disusun sebagai salah satu bentuk penyelesaian proyek akhir semester 1.

Program kami, "Sistem Manajemen Toko Hijab," dirancang untuk membantu pengelolaan penjualan di toko hijab dengan fitur pencatatan nama produk, warna, stok, dan harga, serta mendukung proses pembelian oleh pelanggan.

Selama penyusunan program, kami menghadapi kendala seperti pemahaman materi pemrograman yang kompleks, manajemen waktu juga menjadi tantangan karena harus membagi fokus antara perkuliahan dan proyek akhir dan pembagian tugas dalam pengerjaan. Semua kendala ini berhasil diatasi berkat dukungan dari asisten praktikum, rekan diskusi, dan kerja sama tim. Dan tidak lupa dalam penyusunan laporan ini kami mengucapkan terimakasih yang sebesar besarnya pada Bang Celio Arga selaku pembimbing kelompok kami yang telah memberikan bimbingan dan arahan. Para asisten Laboratorium Informatika yang mendukung selama pembelajaran. Seluruh teman teman yang telah bekerjasama dalam menyelesaikan Proyek Akhir ini.

Kami menyadari bahwa laporan ini masih memiliki keterbatasan dan ketidaksempurnaan. Oleh karena itu, kami mengharapkan kritik dan saran yang membangun demi penyempurnaan laporan ini. Semoga laporan ini dapat bermanfaat bagi semua pihak yang membacanya. Terima kasih

Samarinda, 13 November 2024

Kelompok 6

Kelompok 6 A1 2024 |ii

TAKARIR

<i>Print</i>	Mencetak
<i>Input</i>	Memasukkan
<i>Read</i>	Membaca
<i>Clear</i>	Membersihkan
<i>System</i>	Sistem
<i>Open</i>	Membuka
<i>New line</i>	Baris baru
<i>Writer</i>	Menulis
<i>DataFrame</i>	Kerangka data
<i>Item</i>	Barang
<i>True</i>	Benar
<i>False</i>	Salah
<i>Users</i>	Pengguna
<i>Username</i>	Nama belakang
<i>Password</i>	Kata sandi
<i>Role</i>	Peran
<i>Append</i>	Menambahkan
<i>Not</i>	Bukan
<i>Or</i>	Atau
<i>And</i>	Dan
<i>Return</i>	Kembali
<i>Order</i>	Memesan
<i>Remove</i>	Menghapus
<i>Import</i>	Impor
<i>Entry</i>	Pintu masuk
<i>Blue</i>	Biru
<i>For</i>	Untuk
<i>If</i>	Jika

DAFTAR ISI

KATA PENGANTAR	ii
TAKARIR	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	v
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Kebutuhan Fungsional.....	4
1.3 Rumusan Masalah	4
1.4 Batasan Masalah.....	5
1.5 Tujuan.....	5
BAB II PERENCANAAN	6
2.1 Analisis Program	6
2.2 Flowchart.....	7
1.1 Konsep/Materi Praktikum	18
BAB III HASIL DAN PEMBAHASAN	23
3.1 Tampilan Program	23
3.2 Source Code	37
BAB IV PENUTUP	51
4.1 Kesimpulan.....	51
4.2 Saran	51
DAFTAR PUSTAKA	52
LAMPIRAN	53

DAFTAR GAMBAR

Gambar 2. 1. 1 Menu Utama	7
Gambar 2. 1. 2 Menu Registrasi.....	8
Gambar 2. 1. 3 Menu Login	9
Gambar 2. 1. 4 Menu Admin	10
Gambar 2. 1. 5 Menu Tambahkan Produk.....	11
Gambar 2. 1. 6 Menu Tampilkan Produk.....	12
Gambar 2. 1. 7 Menu Tambah Akun Admin	12
Gambar 2. 1. 8 Menu Tampilkan Pengunjung.....	13
Gambar 2. 1. 9 Menu Hapus Akun Pengunjung.....	14
Gambar 2. 1. 10 Menu Ubah Produk	15
Gambar 2. 1. 11 Menu Hapus Produk.....	16
Gambar 2. 1. 12 Menu Pengguna Biasa.....	17
Gambar 2. 1. 13 Menu Buat Pesanan	18
Gambar 3. 1. 1 Tampilan Menu Utama.....	23
Gambar 3. 1. 2 Tampilkan Menu Registrasi Sebagai Pengguna Biasa.....	23
Gambar 3. 1. 3 Tampilan Menu Login Sebagai Pengguna Biasa.....	24
Gambar 3. 1. 4 Tampilan Menu Menampilkan Produk.....	25
Gambar 3. 1. 5 Tampilan Menu Membuat Pesanan	26
Gambar 3. 1. 6 Tampilan Menu Kritik/Saran	27
Gambar 3. 1. 7 Tampilan Menu Login Sebagai Admin.....	28
Gambar 3. 1. 8 Tampilan Menu Tambahkan Produk	29
Gambar 3. 1. 9 Tampilan Menu Menampilkan Produk.....	30
Gambar 3. 1. 10 Tampilan Menu Menambahkan Akun Admin Baru.....	31
Gambar 3. 1. 11 Tampilan Menu Menampilkan Daftar Pengunjung	32
Gambar 3. 1. 12 Tampilan Menu Menghapus Akun Pengguna Biasa.....	33
Gambar 3. 1. 13 Tampilan Menu Mengubah Produk.....	34
Gambar 3. 1. 14 Tampilan Menu Menghapus Produk.....	35
Gambar 3. 1. 15 Tampilan Menu Lihat Kritik/Saran	36

Gambar 1 Cover Kartu Konsul.....	53
Gambar 2 Kartu Konsultasi Pertama	54
Gambar 3 Kartu Konsultasi Kedua.....	55

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia usaha, terutama di sektor ritel seperti toko hijab, pengelolaan data dan informasi yang efisien sangat penting untuk memastikan kelancaran operasional dan kepuasan pelanggan. Namun, banyak toko hijab yang masih menggunakan metode manual atau tradisional dalam mengelola data produk, transaksi, dan pengguna. Hal ini menyebabkan sejumlah masalah yang dapat menghambat kinerja dan perkembangan toko, antara lain:

1. Pengelolaan Data Produk yang Tidak Efisien

Banyak toko hijab yang masih menggunakan sistem manual atau spreadsheet sederhana dalam mengelola data produk hijab, seperti nama produk, harga, stok, dan warna. Pengelolaan data produk yang tidak terstruktur dengan baik ini dapat menyebabkan kesalahan dalam pencatatan stok, informasi produk yang tidak up-to-date, dan kebingungan saat admin perlu mencari data produk tertentu. Kesulitan ini semakin diperburuk ketika stok produk berubah dengan cepat dan sering.

2. Proses Pemesanan yang Rumit

Bagi pelanggan, proses pemesanan produk hijab di toko fisik atau online yang tidak sistematis sering kali mempersulit pengalaman berbelanja. Jika pelanggan harus menghubungi pemilik toko secara manual (misalnya, lewat telepon atau WhatsApp) untuk melakukan pemesanan, hal ini bisa memakan waktu dan berpotensi menyebabkan kesalahan dalam mencatat jumlah atau jenis produk yang dipesan. Selain itu, tidak ada sistem untuk melacak pesanan, sehingga membuat proses pemesanan menjadi tidak transparan.

3. Kesulitan dalam Pengelolaan Pengguna dan Akun

Dalam toko yang sudah memiliki banyak pelanggan, kesulitan dapat muncul dalam mengelola akun pengguna, seperti dalam hal pendaftaran, login, serta perubahan data pengguna. Sistem manual atau tidak terintegrasi dapat mengakibatkan kesalahan pencatatan data akun, mempersulit pendaftaran pengguna baru, serta menghambat pengelolaan akun pengguna seperti penutupan akun atau pemberian hak akses berbeda (misalnya antara pengguna biasa dan admin).

4. Kurangnya Feedback dari Pengguna

Banyak toko yang tidak memiliki mekanisme formal untuk menerima kritik atau saran dari pelanggan. Padahal, umpan balik dari pelanggan adalah sumber informasi yang sangat berharga untuk meningkatkan kualitas produk dan layanan. Tanpa adanya sistem untuk menampung dan mengelola kritik atau saran, toko akan kesulitan dalam mengetahui area yang perlu diperbaiki, baik itu terkait dengan produk atau pelayanan.

5. Pengelolaan Transaksi yang Tidak Tersentralisasi

Banyak toko yang menggunakan sistem manual dalam mencatat transaksi pemesanan, yang berpotensi menyebabkan kesalahan dalam perhitungan total harga, stok produk, serta kesulitan dalam memantau riwayat pesanan. Tanpa sistem yang memadai, pemilik toko akan kesulitan untuk memantau berapa banyak produk yang terjual, siapa yang memesan, dan apa saja produk yang laris terjual.

6. Kesulitan dalam Penyimpanan dan Pengolahan Data

Sistem yang tidak terorganisir menyebabkan kesulitan dalam penyimpanan dan pengelolaan data, seperti data produk, pengguna, pesanan, dan kritik/saran. Ketika data disimpan dalam berbagai format

yang tidak terstruktur atau di tempat yang berbeda-beda, akan ada masalah dalam pengambilan dan pembaruan data yang cepat dan akurat.

7. Keterbatasan Akses dan Keamanan Data

Tanpa sistem yang terintegrasi, data yang ada dapat terancam kehilangan atau kerusakan. Selain itu, dalam sistem manual, keamanan data pengguna (seperti username dan password) juga seringkali kurang terjamin. Tanpa proteksi yang memadai, data sensitif pelanggan atau admin bisa saja jatuh ke tangan yang tidak berwenang.

8. Keterbatasan dalam Skalabilitas Operasional

Seiring dengan berkembangnya bisnis dan meningkatnya jumlah produk serta pesanan, sistem manual yang digunakan akan menjadi semakin tidak efisien dan sulit dikelola. Toko yang menggunakan cara manual akan kesulitan untuk mengatasi lonjakan pesanan, pengelolaan stok yang lebih besar, dan pencatatan transaksi yang lebih kompleks.

Dengan adanya berbagai permasalahan tersebut, toko hijab membutuhkan sebuah sistem yang dapat membantu mengatasi tantangan ini. Sistem berbasis komputer yang terstruktur, mudah digunakan, dan memiliki fitur-fitur yang sesuai dengan kebutuhan operasional toko, seperti pengelolaan produk, pesanan, akun pengguna, kritik dan saran, serta laporan transaksi, akan sangat bermanfaat. Implementasi sistem yang baik tidak hanya akan meningkatkan efisiensi pengelolaan, tetapi juga memberikan pengalaman yang lebih baik bagi pelanggan dan admin dalam menjalankan toko hijab.

1.2 Kebutuhan Fungsional

Sistem pengelolaan toko hijab ini memiliki beberapa kegunaan fungsional utama untuk mendukung operasional toko.

1. Sistem memudahkan manajemen produk, mulai dari penambahan, pengeditan, hingga penghapusan produk dengan data yang selalu terbaru.
2. Sistem memungkinkan pengelolaan akun pengguna dengan dua kategori: pengguna biasa dan admin, di mana admin memiliki akses penuh untuk mengelola data pengguna.
3. Proses pemesanan produk juga diatur dengan mudah, termasuk pengurangan stok otomatis sesuai jumlah pemesanan.
4. Data transaksi dan feedback pelanggan disimpan dengan baik untuk memudahkan pelaporan dan analisis produk.
5. Menyediakan fitur keamanan dengan kontrol akses yang membatasi hak admin dan pengguna biasa, serta memudahkan pemantauan stok barang.

Secara keseluruhan, sistem ini meningkatkan efisiensi operasional, mempermudah pengelolaan produk, dan membantu meningkatkan kepuasan pelanggan.

1.3 Rumusan Masalah

Beberapa rumusan masalah yang dapat diidentifikasi dalam pengembangan sistem ini antara lain:

1. Bagaimana cara mengelola data produk hijab secara efektif?
2. Bagaimana cara mengelola data pengguna dengan aman dan terstruktur?
3. Bagaimana cara menangani pesanan dari pengguna dengan sistem yang efisien?
4. Bagaimana cara mengelola kritik dan saran dari pengguna untuk meningkatkan kualitas layanan?
5. Bagaimana cara mengelola akun pengunjung atau pengguna dengan aman?

6. Bagaimana memastikan sistem selalu terupdate dengan data terbaru?

1.4 Batasan Masalah

Untuk menjaga agar pengembangan sistem ini tetap terfokus dan terkelola dengan baik, beberapa batasan masalah perlu ditetapkan. Adapun batasan masalah dalam pengembangan sistem manajemen toko hijab adalah sebagai berikut:

1. Sistem ini masih berupa program saja dan belum dapat diterapkan sepenuhnya dalam kehidupan nyata.
2. Sistem ini hanya berfokus pada manajemen toko hijab saja.
3. Pengelolaan data menggunakan file CSV.
4. Sistem ini hanya memiliki 2 role yaitu pengguna dan admin.
5. Sistem merupakan program sederhana.
6. Sistem ini hanya dapat melakukan proses pemesanan sederhana
7. Fitur keamanan yang terbatas.
8. Tidak ada sistem backup otomatis.
9. Sistem ini dibuat menggunakan bahasa pemrograman Python.

1.5 Tujuan

Adapun tujuan yang dapat dipaparkan dari penyusunan dan pembuatan program ini adalah sebagai berikut :

1. Meningkatkan efisiensi pengelolaan produk hijab.
2. Menyediakan platform transaksi pesanan yang sederhana.
3. Memberikan sarana umpan balik bagi pengguna.
4. Mempermudah pengelolaan data pengguna.

BAB II

PERENCANAAN

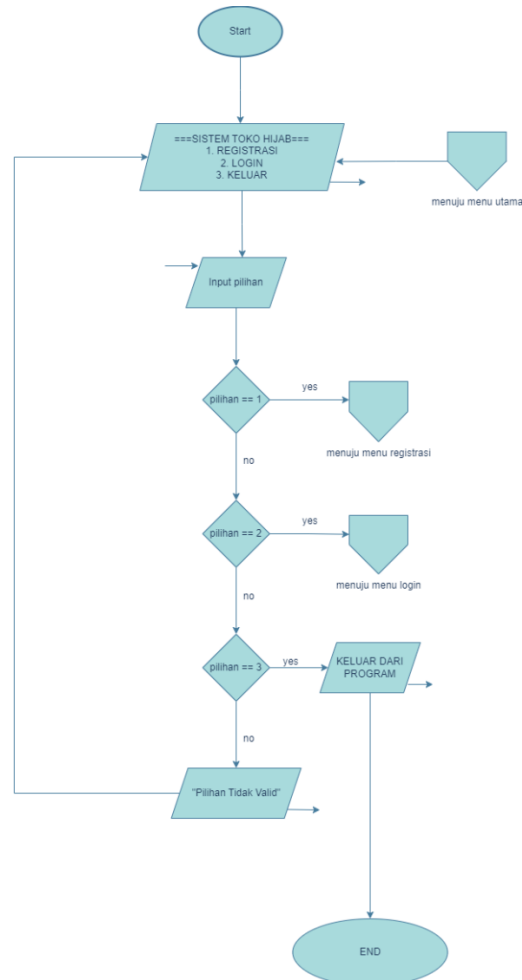
2.1 Analisis Program

Sistem Manajemen Produk Hijab ini dirancang untuk mengelola data hijab dalam sebuah toko dengan menu utama seperti registrasi pengguna, login, dan keluar dari program. Pengguna baru dapat mendaftar dengan username dan password, sementara sistem memastikan bahwa username tersebut belum terdaftar. Setelah login, pengguna dapat diidentifikasi sebagai admin atau pengguna biasa. Admin memiliki kemampuan untuk menambahkan produk, menampilkan produk, menambahkan akun admin, menampilkan daftar pengunjung, menghapus akun pengunjung, menghapus produk, mengubah produk, lihat kritik/saran dan keluar dari menu admin. Sedangkan pengguna biasa hanya dapat melihat daftar produk yang tersedia, melakukan pembelian, memberikan kritik/saran dan keluar dari menu pengguna biasa.

Dengan tampilan yang ramah pengguna. Sistem ini menggunakan loop untuk menjaga agar aplikasi tetap aktif hingga pengguna memilih untuk keluar, serta menampilkan pesan kesalahan jika ada input yang tidak valid, memastikan pengalaman yang lebih baik dalam pengelolaan produk hijab.

2.2 Flowchart

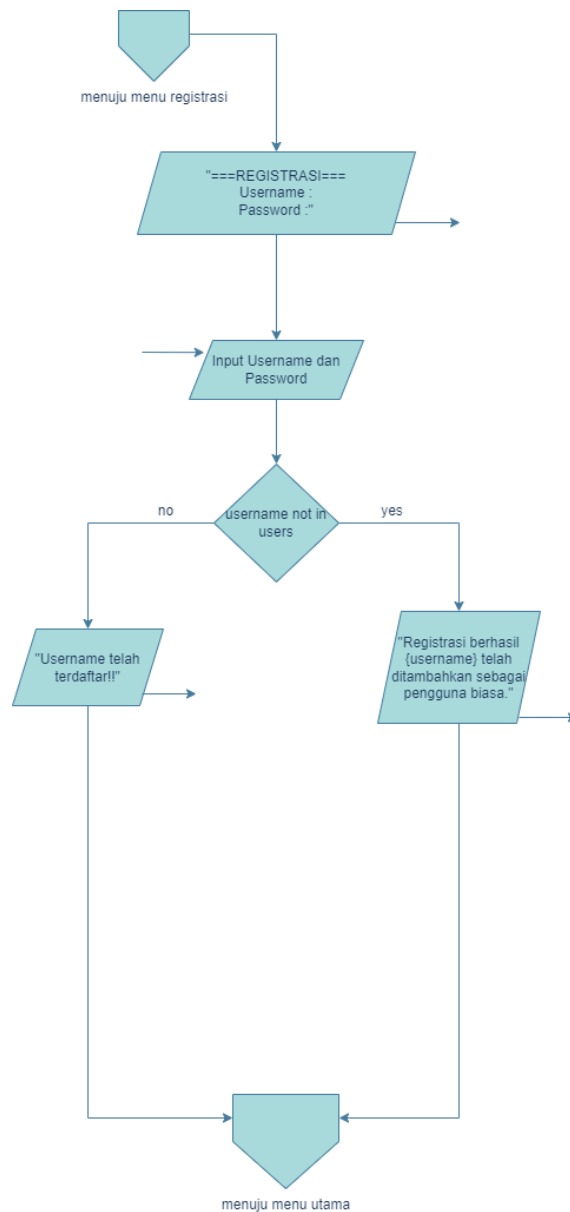
1. Menu Utama



Gambar 2. 1. 1 Menu Utama

Program dimulai dengan menampilkan menu utama yaitu sistem toko hijab. Kemudian pengguna dapat memilih atau menginput menu yang diinginkan. Jika pengguna menginput pilihan 1 maka pengguna akan menuju menu registrasi. Jika pengguna menginput pilihan 2 maka pengguna akan menuju ke menu login dan jika pengguna menginput pilihan 3 maka pengguna akan keluar dari program dan program berhenti. Jika pengguna memasukkan pilihan selain 1-3 maka pengguna akan kembali ke menu utama dan diminta untuk memasukkan ulang pilihan.

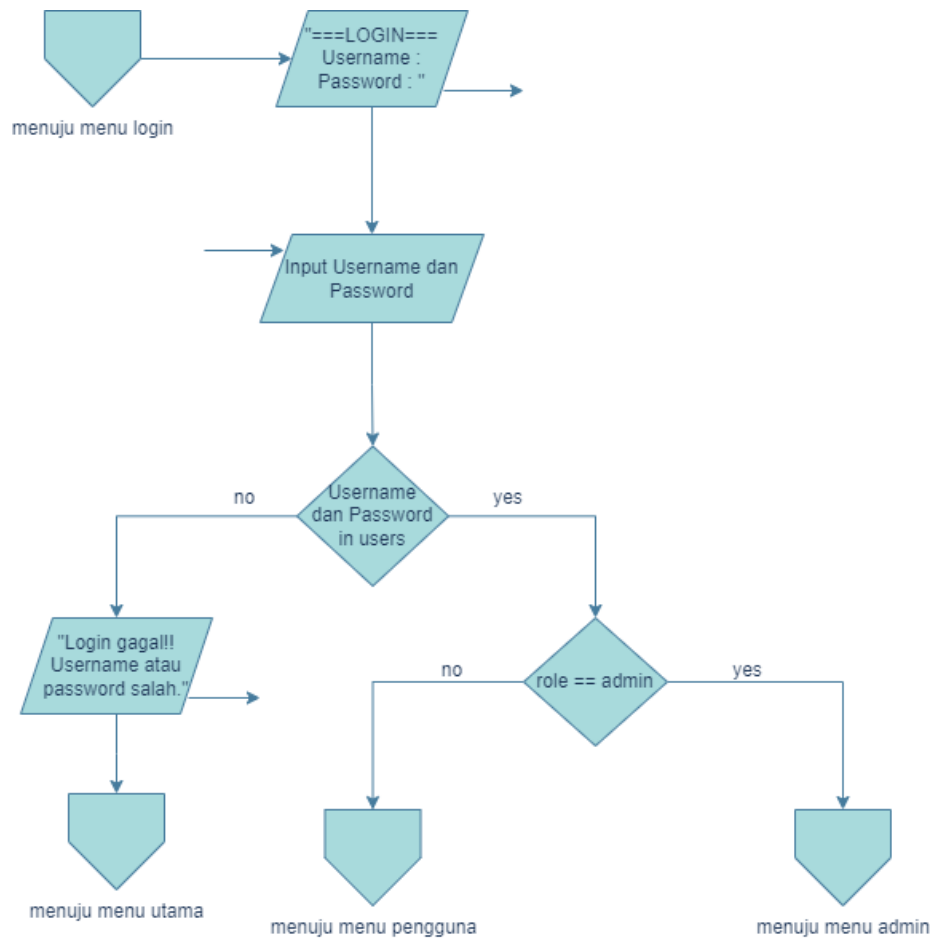
2. Menu Registrasi



Gambar 2. 1. 2 Menu Registrasi

Pada saat pengguna berada di menu registrasi, pengguna diminta untuk memasukkan username dan password. Kemudian program akan mengecek apakah username sudah ada/terdaftar. Jika sudah terdaftar maka username baru tidak berhasil dibuat dan akan kembali ke menu utama. Jika username belum terdaftar maka username yang baru berhasil ditambahkan dan akan kembali ke menu utama.

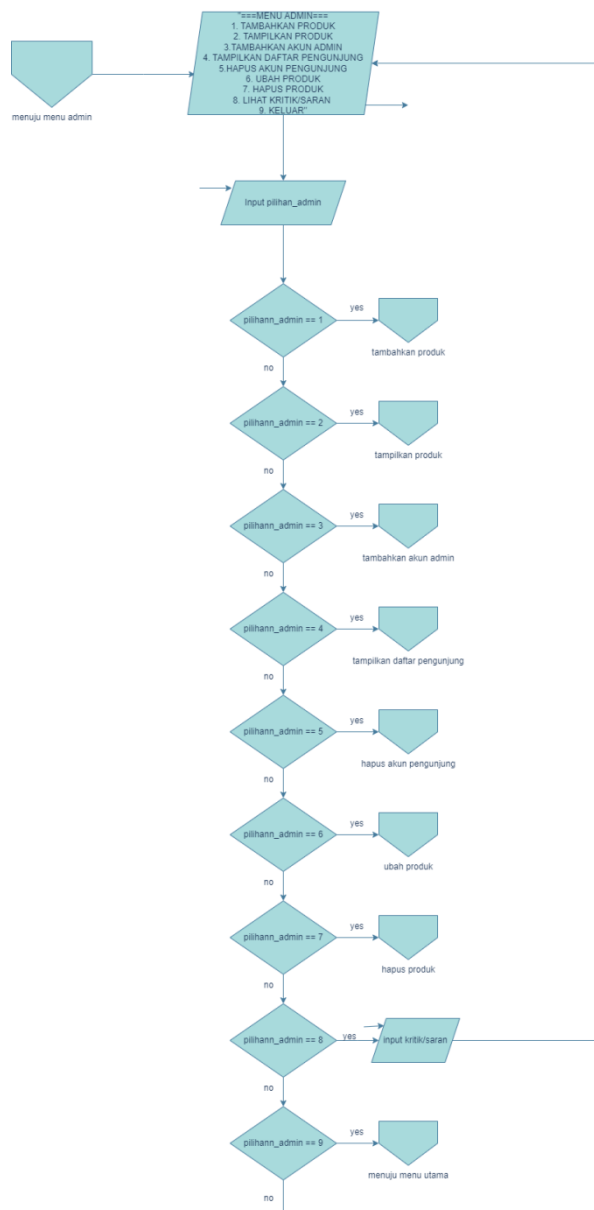
3. Menu Login



Gambar 2. 1. 3 Menu Login

Pada saat pengguna berada di menu login. Pengguna diminta untuk memasukkan username dan password. Jika username dan password tidak cocok maka pengguna tidak dapat login dan akan kembali ke menu utama namun, jika username dan password cocok maka dapat login sesuai dengan rolenya. Jika rolenya admin maka akan menuju ke menu admin dan jika rolenya pengguna biasa maka akan menuju ke menu pengguna biasa.

4. Menu Admin

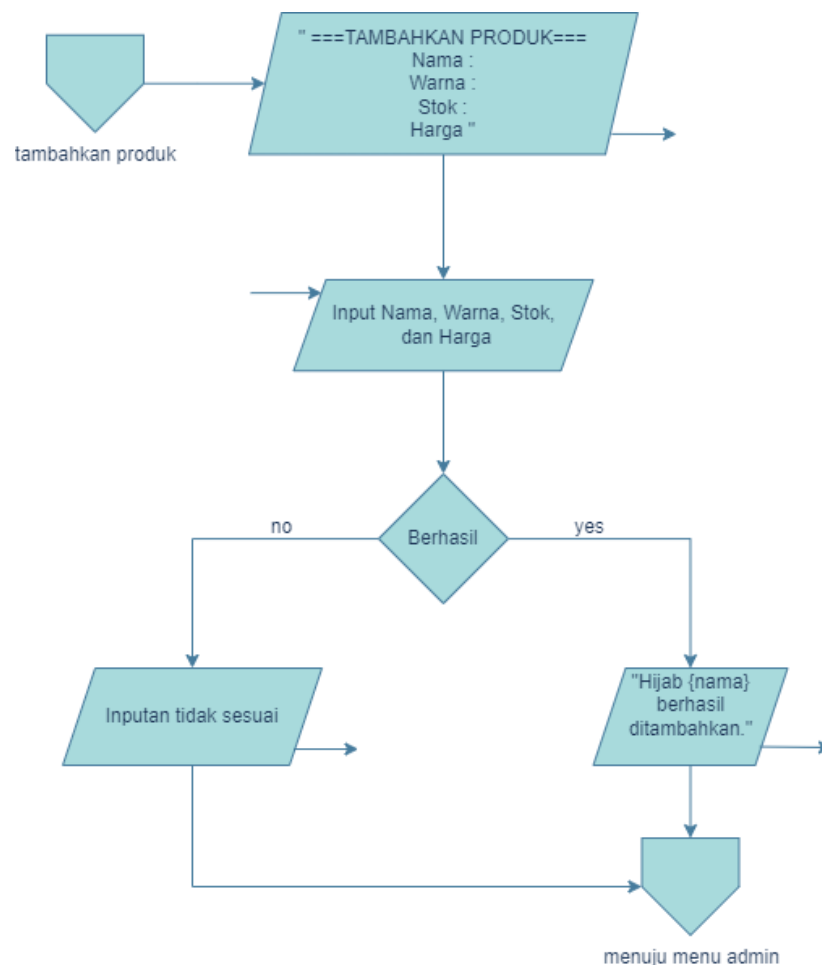


Gambar 2. 1. 4 Menu Admin

Pada saat berada di menu admin. Admin di minta untuk memasukkan pilihan. Jika memasukkan pilihan 1 admin menuju menu tambahkan produk. Jika memasukkan pilihan 2 admin menuju menu tampilkan produk. Jika memasukkan pilihan 3 admin menuju menu tambah akun admin. Jika memasukkan pilihan 4 admin menuju menu tampilkan daftar pengunjung. Jika memasukkan pilihan 5 admin menuju

menu hapus akun pengunjung. Jika memasukkan pilihan 6 admin menuju menu ubah produk. Jika memasukkan pilihan 7 admin menuju menu hapus produk. Jika memasukkan pilihan 8 admin menuju menu kritik/saran. Jika memasukan pilihan selain angka 1-8 maka admin akan dikembali ke menu admin untuk memasukkan pilihan lagi.

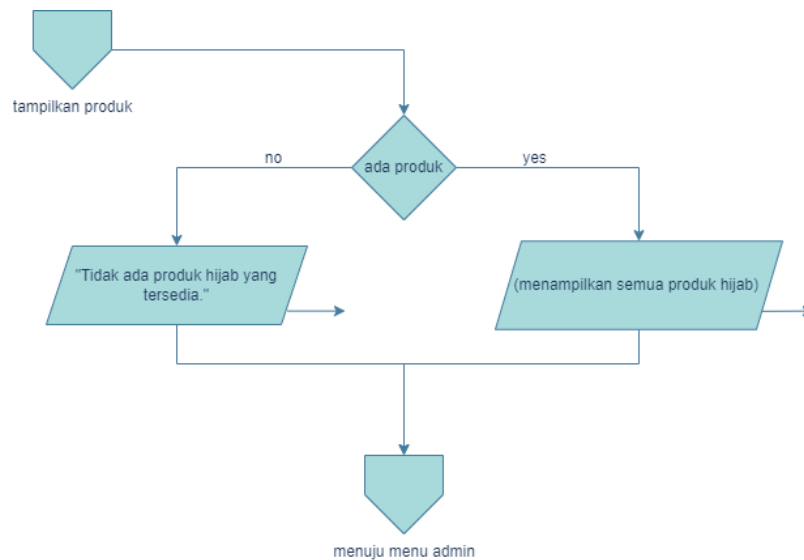
5. Menu Tambahkan Produk



Gambar 2. 1. 5 Menu Tambahkan Produk

Pada saat berada di menu tambahkan produk. Admin diminta untuk memasukkan nama, warna, stok, dan harga. Jika inputan valid maka produk berhasil ditambahkan dan kembali ke menu admin namun, jika tidak valid maka produk tidak berhasil ditambahkan dan admin akan kembali ke menu admin.

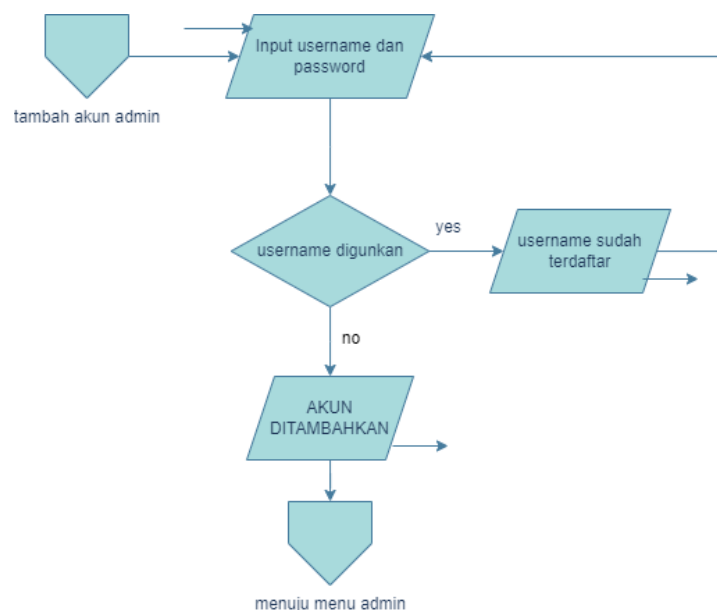
6. Menu Tampilkan Produk



Gambar 2. 1. 6 Menu Tampilkan Produk

Pada saat berada di menu tampilkan produk. Jika terdapat produk maka akan menampilkan semua produk hijab dan akan kembali ke menu admin. Namun jika tidak ada produk maka akan menampilkan pesan dan akan kembali ke menu admin.

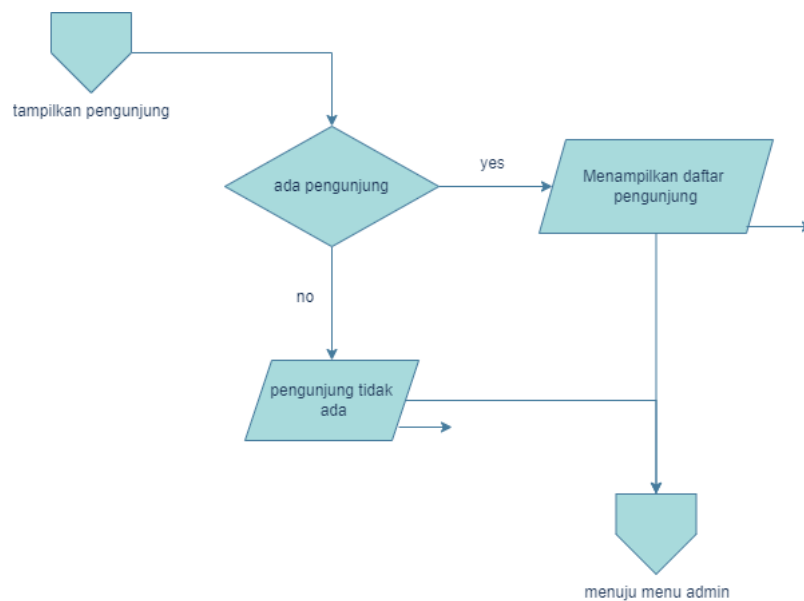
7. Menu Tambah Akun Admin



Gambar 2. 1. 7 Menu Tambah Akun Admin

Pada saat berada di menu tambah akun admin maka admin di minta untuk memasukkan username dan password. Jika username telah digunakan maka akan diminta ulang untuk mencoba lagi. Jika username belum digunakan maka username baru berhasil ditambahkan dan akan kembali ke menu admin.

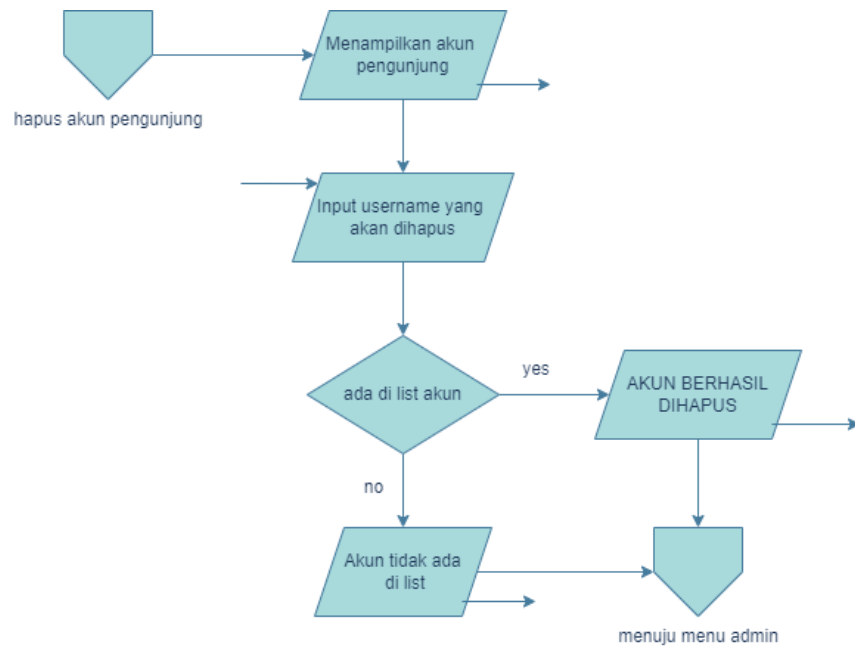
8. Menu Tampilkan Pengunjung



Gambar 2. 1. 8 Menu Tampilkan Pengunjung

Pada saat berada di menu tampilkan pengunjung. Jika terdapat pengunjung maka akan menampilkan semua pengunjung dan akan kembali ke menu admin. Namun jika tidak ada pengunjung maka akan menampilkan pesan dan akan kembali ke menu admin.

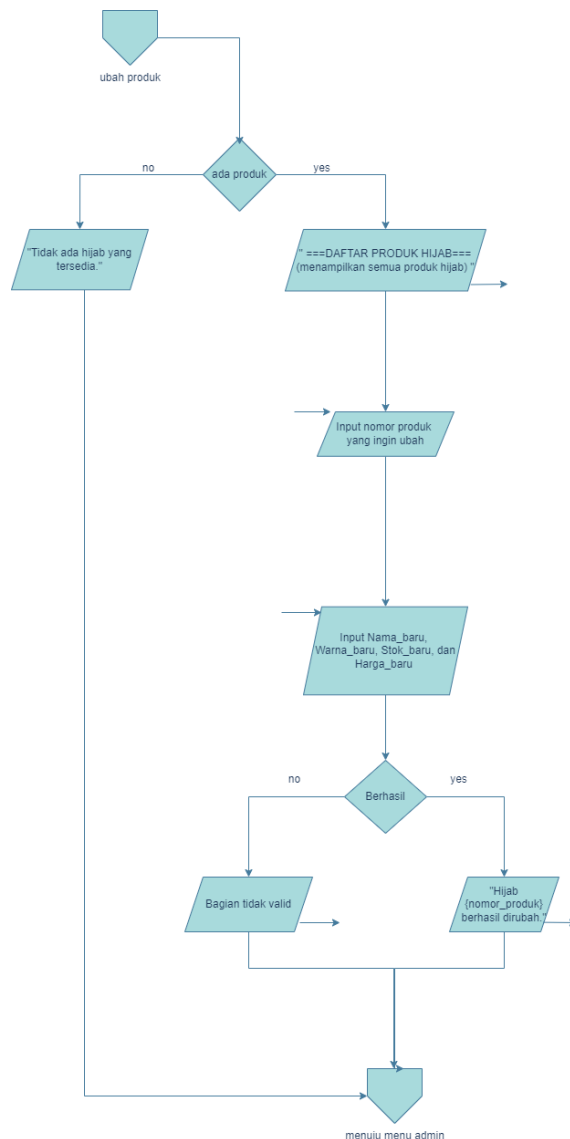
9. Menu Hapus Akun Pengunjung



Gambar 2. 1. 9 Menu Hapus Akun Pengunjung

Pada saat berada di menu hapus akun pengunjung. Program akan menampilkan semua akun pengunjung dan meminta admin memasukkan no akun yang ingin dihapus jika valid maka akun akan dihapus dan akan kembali ke menuju menu admin namun, jika tidak valid maka akan diminta untuk mencoba memasukkan no akun lagi.

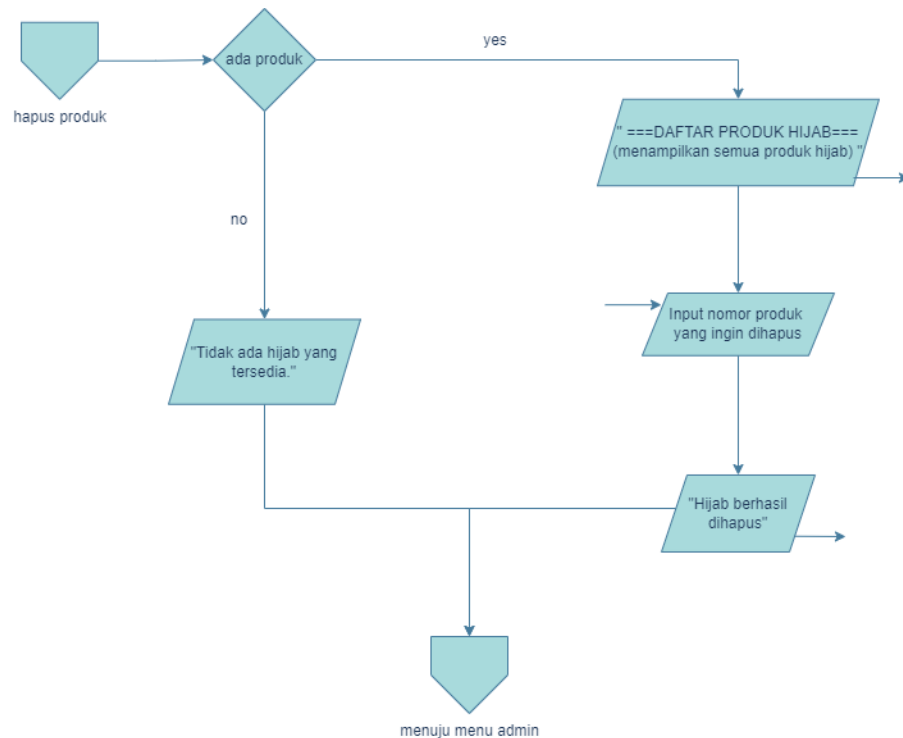
10. Menu Ubah Produk



Gambar 2. 1. 10 Menu Ubah Produk

Pada saat berada di menu ubah produk. Jika tidak ada produk program akan menampilkan pesan dan akan kembali menuju menu admin. Jika produk ada maka program akan menampilkan semua produk dan admin di minta untuk memasukkan nomor produk yang akan diubah, jika valid maka admin akan memasukkan nama baru, warna baru, stok baru dan harga baru jika valid maka produk akan disimpan dan akan ke menu admin. Jika nomor produk tidak valid maka akan menampilkan pesan dan kembali ke menu admin.

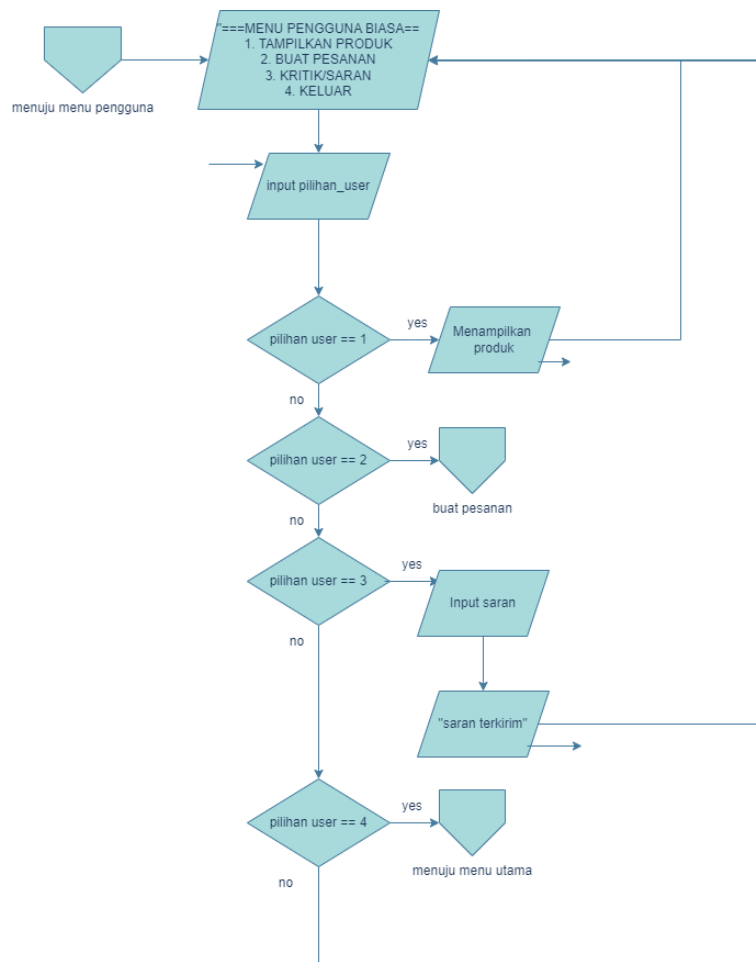
11. Menu Hapus Produk



Gambar 2. 1. 11 Menu Hapus Produk

Pada saat berada di menu hapus produk. Jika ada produk maka program akan menampilkan semua produk dan admin di minta untuk memasukkan nomor produk yang ingin dihapus dan produk pun akan berhasil dihapus dan akan kembali ke menu admin. Jika tidak ada produk maka program akan menampilkan pesan dan akan kembali ke menu admin.

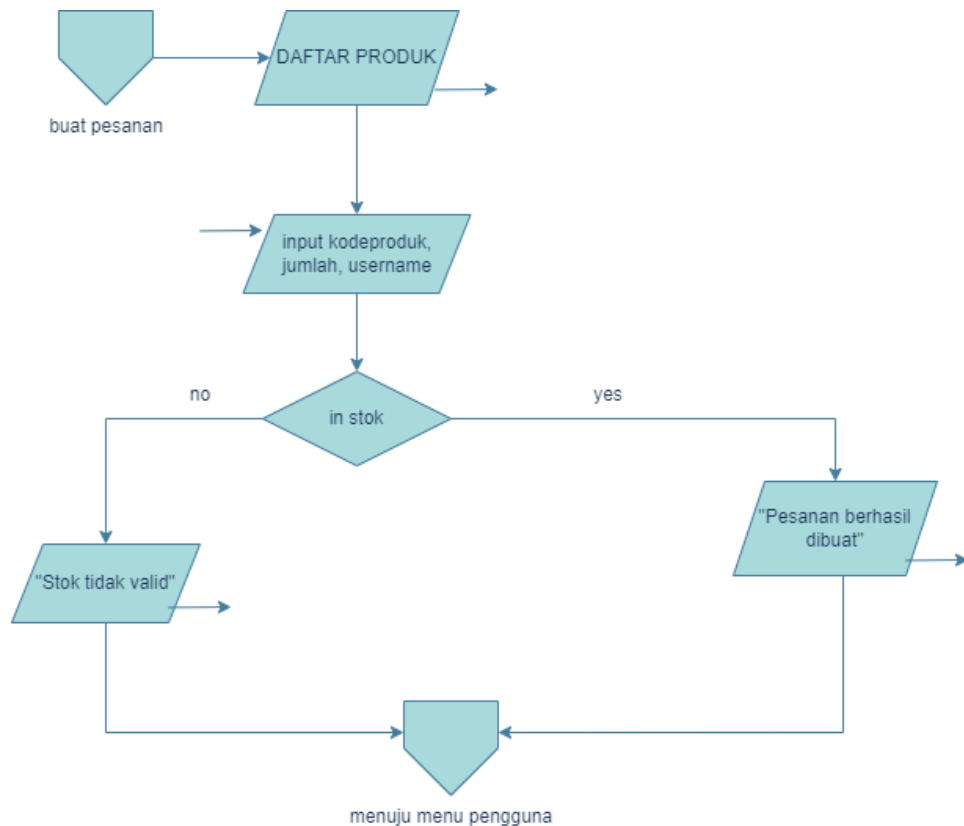
12. Menu Pengguna Biasa



Gambar 2. 1. 12 Menu Pengguna Biasa

Pada saat berada di menu pengguna biasa. Pengguna biasa di minta untuk memasukkan pilihan. Jika memasukkan pilihan 1 maka program akan menampilkan semua produk hijab jika ada dan akan kembali ke menu pengguna biasa. Jika memasukkan pilihan 2 maka akan menuju ke menu pesanan dan akan kembali ke menu pengguna biasa. Jika memasukkan pilihan 3 maka pengguna biasa akan memasukkan saran dan saran dikirim dan akan kembali ke menu pengguna biasa. Jika memasukkan pilihan 4 maka akan keluar dari menu pengguna biasa dan kembali ke menu utama. Jika memasukan pilihan selain angka 1-3 maka pengguna biasa akan dikembali ke menu pengguna biasa dan memasukkan pilihan lagi.

13. Menu Buat Pesanan



Gambar 2. 1. 13 Menu Buat Pesanan

Pada saat berada di menu pesanan. Program akan menampilkan daftar produk. Pengguna diminta untuk memasukkan kode produk dan jumlah. Jika ada stok maka pesanan dibuat dan kembali ke menu pengguna. Jika tidak ada stok maka pengguna akan kembali ke menu pengguna.

1.1 Konsep/Materi Praktikum

Konsep yang kami gunakan dalam pembuatan laporan ini sesuai dengan persyaratan yang terdapat pada modul diantaranya:

1. Fungsi Dasar

Fungsi dasar yang kami pakai dalam program kami adalah fungsi print. Fungsi print disebut juga sebagai fungsi dasar yang kami gunakan dalam program kami, fungsi print program merupakan fungsi

yang paling umum dipakai untuk menampilkan suatu keluaran pada layar peraga.

2. Fungsi Number (int)

Fungsi number adalah tipe data number merepresentasikan nilai-nilai berupa angka. Tipe data yang semua nilainya berupa angka, yang bisa ditambah, dikurang, dikali, dibagi, dan sebagainya. Pada python, tipe data numerik ini dibagi lagi. Yaitu Interger dan Float.

a. Interger

Tipe data yang memiliki nilai bilangan bulat saja. Semua yang bernilai bulat adalah tipe data interger.

b. Float

Tipe data yang seperti interger, namun dia menampung nilai-nilai pecahan/decimal

3. Fungsi String (str)

Tipe data selanjutnya adalah tipe data string. String merupakan tipe data yang menyimpan sebuah teks. Tipe data string ini harus diapit oleh tanda kutip. Baik tanda kutip satu (' ') atau tanda kutip dua (" ").

4. Fungsi CRUD

Aplikasi create, read, update dan delete (CRUD) merupakan operasi dasar yang umum digunakan pada pemrograman database.

5. Fungsi Boolean

Tipe data yang paling mudah dan simple. Namun memiliki peran yang sangat penting untuk membangun program/aplikasi. Tipe data ini hanya memiliki nilai True dan False.

6. Fungsi List

List dapat menyimpan berbagai data sekaligus seperti string, integer, float, dan sebagainya. Kita bahkan dapat menyimpan list di dalam sebuah list yang disebut dengan nested list atau list multi-dimensi. List didefinisikan menggunakan tanda kurung siku [] dan di setiap anggota dari data tersebut dipisahkan oleh tanda koma.

7. Fungsi Append (item)

Fungsi append (item) adalah sebuah fungsi dalam python yang digunakan untuk menambahkan item dari belakang dan menambahkan objek-objek ke list.

8. Fungsi Dictionary

Fungsi Dictionary adalah sebuah tipe data yang anggotanya terdiri dari pasangan kunci : nilai (key :value) untuk merujuk pada isi variabelnya. Dictionary sifat kedua jenis data ini hanya berbeda dalam beberapa hal saja. Untuk mendeklarasikan sebuah dictionary, dalam python menggunakan atau memakai tanda { }.

9. Fungsi Percabangan

Percabangan sering disebut juga sebagai control flow, decision, kondisi atau bisa dibilang sebagai struktur if else. Struktur percabangan dibuat dengan tujuan menentukan tindakan dan perintah sesuai logika/kondisi yang kita berikan.

a. Percabangan If/Else

Percabangan If/Else digunakan saat terdapat dua pilihan keputusan.

b. Percabangan IF / ELIF / ELSE

Percabangan IF / ELIF / ELSE digunakan jika di dalam decision terdapat lebih dari dua pilihan keputusan.

10. Fungsi Perulangan

Perulangan dalam bahasa pemrograman merupakan instruksi yang digunakan untuk memerintahkan komputer melakukan sesuatu secara berulang-ulang dengan jumlah yang ditentukan atau selama masih memenuhi suatu kondisi yang telah ditentukan. Dalam bahasa pemrograman Python terdapat dua macam perulangan, yaitu perulangan for dan perulangan while.

a. Fungsi perulangan For.

Fungsi perulangan For atau yang disebut juga counted loop (perulangan yang terhitung). Perulangan For biasanya digunakan untuk mengulangi kode yang sudah diketahui banyak perulangan. Variabel I berfungsi untuk menampung indeks, dan fungsi range () berfungsi untuk membuat list dengan range.

b. Fungsi perulangan while

Fungsi perulangan while yang disebut uncounted loop (perulangan yang tak terhitung). Perulangan while adalah perulangan yang memiliki syarat dan tidak tentu berapa banyak perulangannya

14. Fungsi Input

Input merujuk pada data atau informasi yang dimasukkan ke dalam sebuah sistem, program, atau perangkat untuk diproses atau digunakan lebih lanjut. Dalam konteks pemrograman, input adalah segala jenis data yang diberikan oleh pengguna atau sumber lain yang kemudian diproses oleh program komputer.

15. Fungsi Output

Fungsi output pada algoritma harus memiliki satu atau lebih output. Suatu algoritma yang tidak memiliki output adalah algoritma yang sia-sia karena algoritma dibuat untuk menghasilkan sesuatu yang diinginkan, yaitu hasil berupa keluaran.

16. Fungsi Def

Fungsi def adalah sebuah blok kode yang terorganisir dengan baik sehingga suatu saat kita dapat menggunakannya kembali (reusable). Atau suatu cara untuk mendefinisikan sebuah method atau fungsi.

17. Fungsi Import

Fungsi import berfungsi untuk multi file maksudnya yaitu kita dapat memanggil atau mengakses file lain didalam satu module yang berbeda.

18. Fungsi Break

Fungsi break adalah perintah khusus yang dipakai untuk memaksa sebuah perulangan berhenti sebelum waktunya. Perintah break bisa digunakan untuk jenis perulangan baik while maupun perulangan for pada bahasa pemrograman Python.

19. Fungsi Pass

Fungsi pass adalah intruksi kepada python bahwa tidak ada kode yang dieksekusi sehingga python melanjutkan mengeksekusi program dibawahnya.

20. Fungsi Return

Fungsi return dalam Python digunakan untuk mengembalikan nilai atau hasil dari sebuah fungsi ke tempat di mana fungsi tersebut dipanggil. Ketika sebuah fungsi dipanggil, eksekusinya akan berhenti pada pernyataan return, dan nilai yang ditentukan setelah kata kunci return akan dikirimkan kembali ke pemanggil fungsi tersebut.

BAB III

HASIL DAN PEMBAHASAN

3.1 Tampilan Program

1. Menu Utama



Gambar 3. 1. 1 Tampilan Menu Utama

Menu tampilan awal terdapat registrasi, login, dan menu keluar. Jadi pada menu registrasi khusus untuk menambahkan akun pengunjung, kemudian pada menu login dapat masuk sebagai pengunjung atau admin, dan yang terakhir menu keluar dari program. Kemudian program meminta untuk memasukkan pilihan.

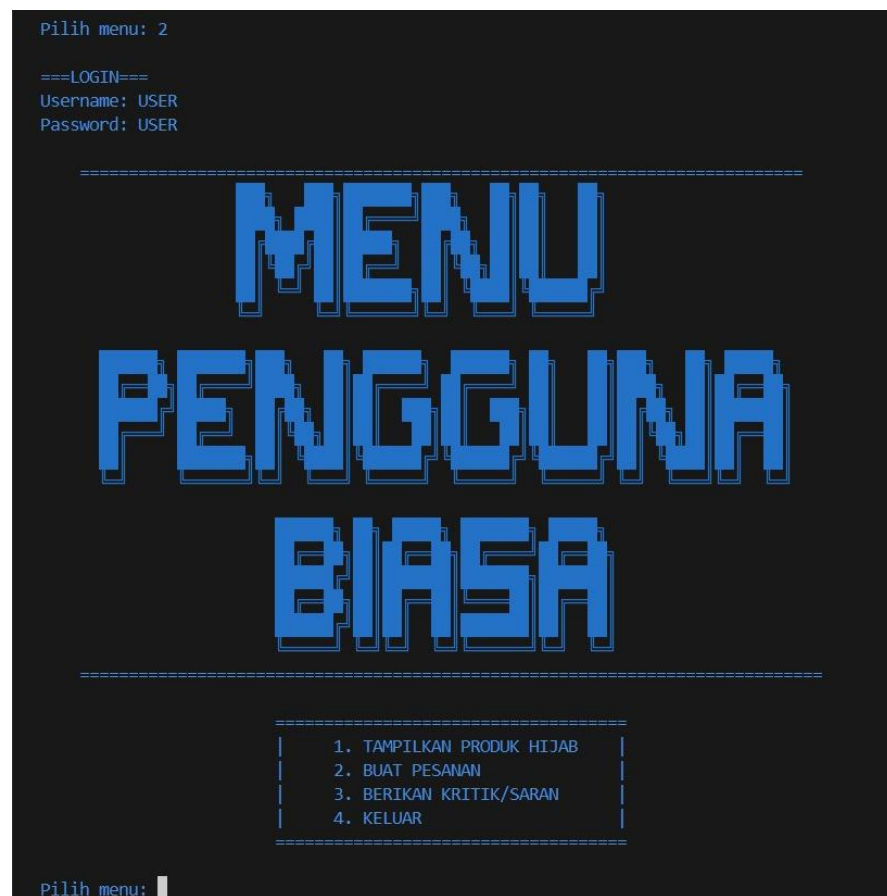
2. Menu Registrasi Sebagai Pengguna Biasa



Gambar 3. 1. 2 Tampilkan Menu Registrasi Sebagai Pengguna Biasa

Pada gambar menampilkan pilihan pertama yaitu registrasi akun pengguna biasa. Program meminta memasukkan username dan password. Dan akan menampilkan pesan berhasil jika berhasil ditambahkan. Kemudian akan kembali ke menu utama

3. Menu Login Sebagai Pengguna Biasa



Gambar 3. 1. 3 Tampilan Menu Login Sebagai Pengguna Biasa

Pada gambar menampilkan login dimana program meminta memasukkan username dan password dan akan diarahkan sesuai rolenya. Pada gambar role sebagai pengguna biasa dan menampilkan menu pengguna biasa. Program pun meminta untuk memasukkan pilihan. Kemudian program meminta untuk memasukkan pilihan.

4. Menu Menampilkan Produk

Kode Produk	Nama	Warna	Stok	Harga (Rp)
1	pahmina	merah	0	20,000
2	Segitiga	Ungu	20	30,000,000
3	Khimar	Hitam	80	10,000,000
4	Bergo	merah	70	50,000,000

MENU PENGGUNA BIASA

1. TAMPILKAN PRODUK HIJAB

2. BUAT PESANAN

3. BERIKAN KRITIK/SARAN

4. KELUAR

Pilih menu:

Gambar 3. 1. 4 Tampilan Menu Menampilkan Produk

Gambar menampilkan produk-produk yang terdapat di toko hijab yang terdiri dari kode produk, nama, warna, stok dan harga dalam rupiah yang dikemas dalam tabel. Kemudian program meminta untuk memasukkan pilihan.

5. Menu Membuat Pesanan

```
+-----+-----+-----+-----+-----+
| Kode Produk | Nama | Warna | Stok | Harga (Rp) |
+-----+-----+-----+-----+-----+
| 1 | pahmina | merah | 0 | 20,000 |
| 2 | Segitiga | Ungu | 20 | 30,000,000 |
| 3 | Khimar | Hitam | 80 | 10,000,000 |
| 4 | Bergo | merah | 70 | 50,000,000 |
+-----+-----+-----+-----+-----+
Masukkan kode produk yang ingin Anda pesan: 3
Masukkan jumlah yang ingin dipesan: 10
Masukkan username Anda: USER
Pesanan berhasil dibuat.

=====
MENU
PENGGUNA
BIASA
=====

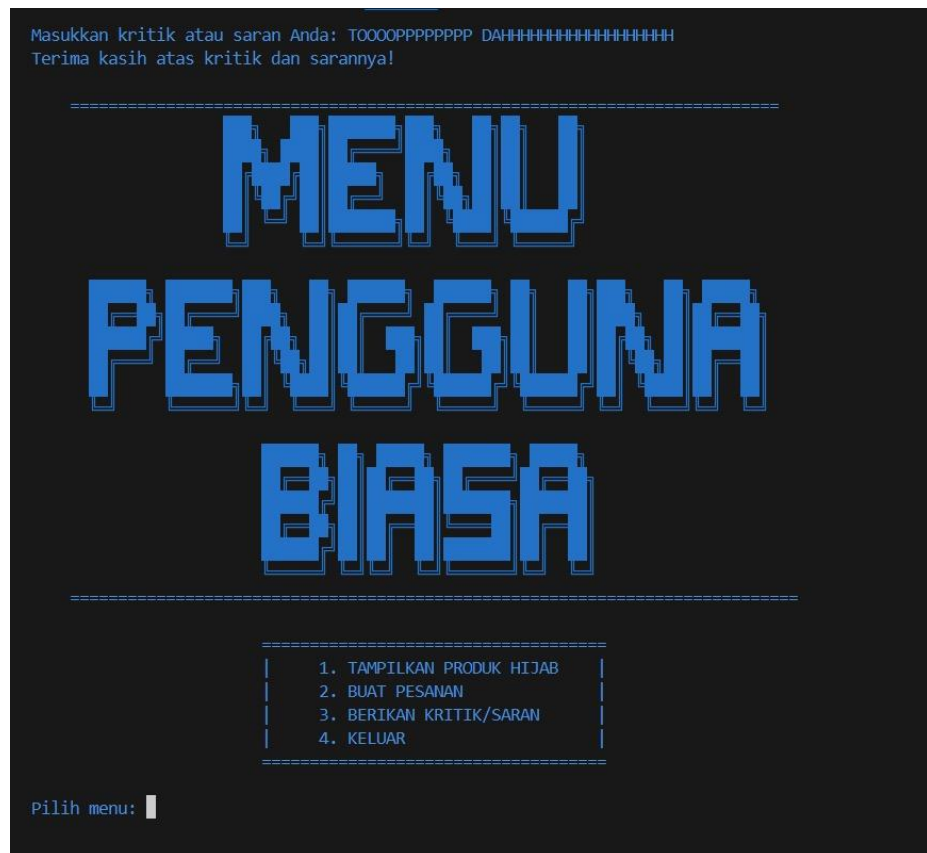
=====
1. TAMPILKAN PRODUK HIJAB
2. BUAT PESANAN
3. BERIKAN KRITIK/SARAN
4. KELUAR
=====

Pilih menu: |
```

Gambar 3. 1. 5 Tampilan Menu Membuat Pesanan

Gambar menampilkan semua produk hijab yang tersedia. Pengguna di minta untuk memasukkan kode produk dan jumlah yang ingin dipesan. Apabila stok produk tersedia maka pesanan berhasil dibuat. Kemudian program meminta untuk memasukkan pilihan.

6. Menu Kritik/Saran



Gambar 3. 1. 6 Tampilan Menu Kritik/Saran

Gambar menampilkan menu kritik atau saran. Pengguna diminta untuk memasukkan kritik atau saran kemudian program akan menampilkan pesan berhasil. Kemudian program meminta untuk memasukkan pilihan.

7. Menu Login Sebagai Admin

```
Pilih menu: 2

===LOGIN===
Username: admin
Password: admin


=====
~~~MENU~~~
=====

ADMIN
=====

=====
|          |
| 1. TAMBAHKAN PRODUK      |
| 2. TAMPILKAN PRODUK     |
| 3. TAMBAHKAN AKUN ADMIN |
| 4. TAMPILKAN DAFTAR PENGUNJUNG |
| 5. HAPUS AKUN PENGUNJUNG |
| 6. UBAH PRODUK          |
| 7. HAPUS PRODUK         |
| 8. LIHAT KRITIK/SARAN   |
| 9. KELUAR               |
|          |
=====
```

Gambar 3. 1. 7 Tampilan Menu Login Sebagai Admin

Pada gambar menampilkan login dimana program meminta memasukkan username dan password dan akan diarahkan sesuai rolenya. Pada gambar role sebagai admin dan menampilkan menu admin. Program pun meminta untuk memasukkan pilihan. Kemudian program meminta untuk memasukkan pilihan.

8. Menu Tambahkan Produk

```
Nama: Scraf
Warna: Jingga
Stok: 100
Harga: 15000000

Hijab Scraf berhasil ditambahkan.

=====
MENU
=====

ADMIN
=====

=====
1. TAMBAHKAN PRODUK
2. TAMPILKAN PRODUK
3. TAMBAHKAN AKUN ADMIN
4. TAMPILKAN DAFTAR PENGUNJUNG
5. HAPUS AKUN PENGUNJUNG
6. UBAH PRODUK
7. HAPUS PRODUK
8. LIHAT KRITIK/SARAN
9. KELUAR
=====

Pilih menu: █
```

Gambar 3. 1. 8 Tampilan Menu Tambahkan Produk

Pada gambar menampilkan menu tambahkan produk dimana admin diminta untuk memasukkan nama, warna, stok, dan harga. Jika inputan valid maka akan menampilkan pesan berhasil. Kemudian program meminta untuk memasukkan pilihan.

9. Menu Menampilkan Produk

Kode Produk	Nama	Warna	Stok	Harga (Rp)
1	pahmina	merah	0	20,000
2	Segitiga	Ungu	20	30,000,000
3	Khimar	Hitam	80	10,000,000
4	Bergo	merah	70	50,000,000
5	Scraf	Jingga	100	15,000,000

MENU

ADMIN

1. TAMBAHKAN PRODUK

2. TAMPILKAN PRODUK

3. TAMBAHKAN AKUN ADMIN

4. TAMPILKAN DAFTAR PENGUNJUNG

5. HAPUS AKUN PENGUNJUNG

6. UBAH PRODUK

7. HAPUS PRODUK

8. LIHAT KRITIK/SARAN

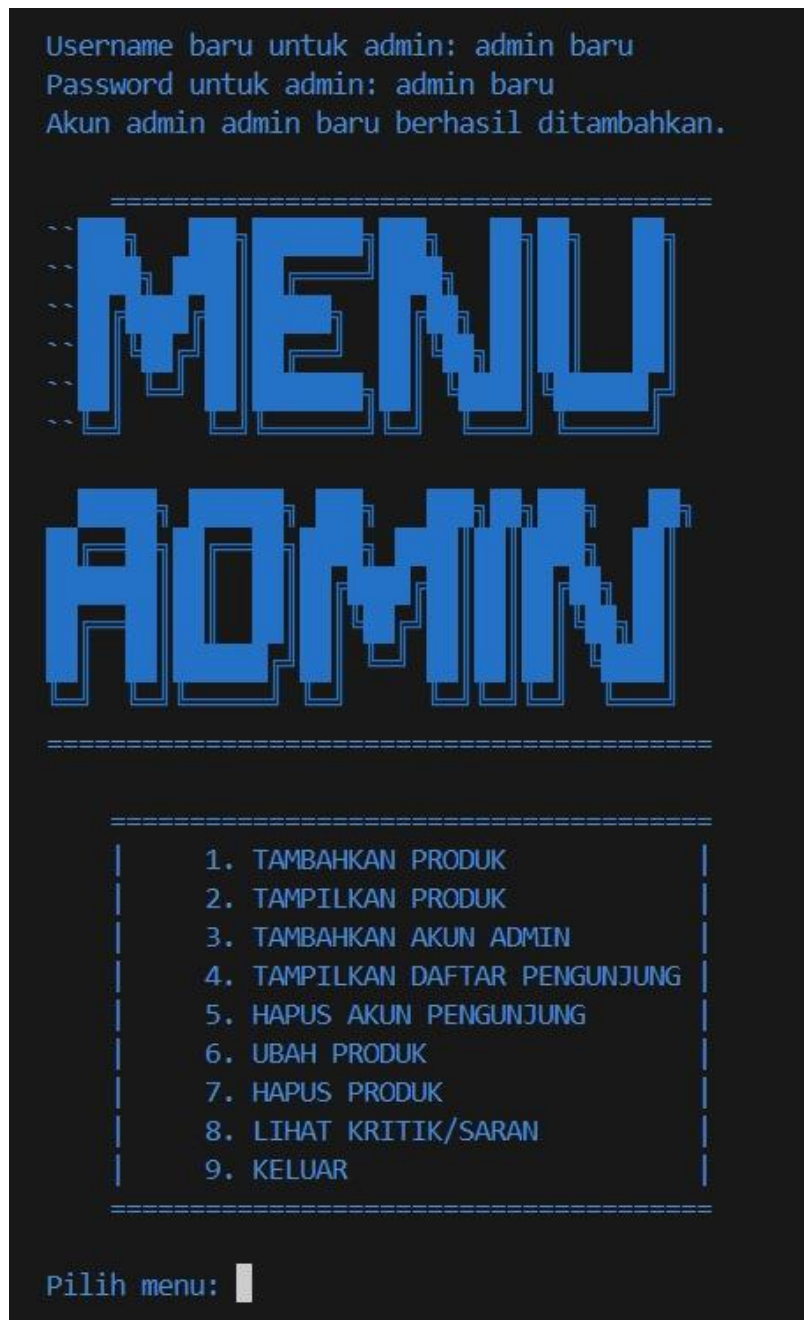
9. KELUAR

Pilih menu:

Gambar 3. 1. 9 Tampilan Menu Menampilkan Produk

Gambar menampilkan produk-produk yang terdapat di toko hijab yang terdiri dari kode produk, nama, warna, stok dan harga dalam rupiah yang dikemas dalam tabel. Kemudian program meminta untuk memasukkan pilihan.

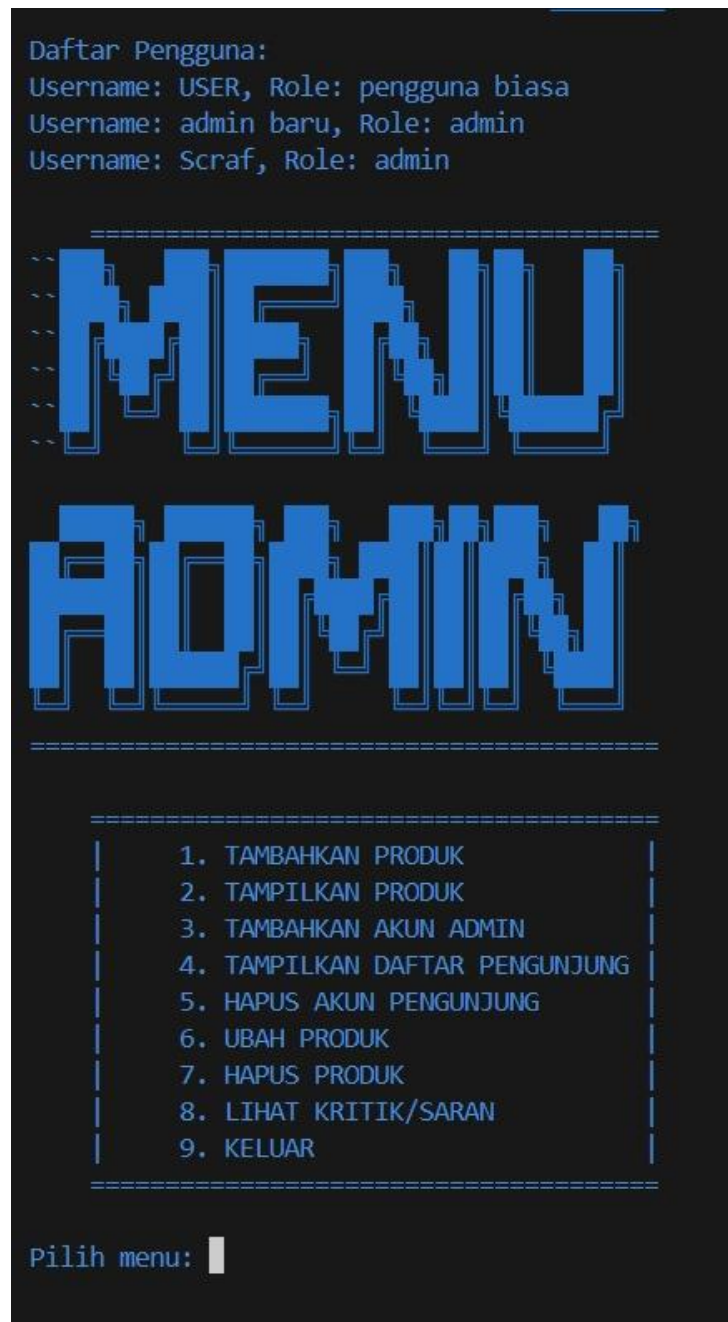
10. Menu Menambahkan Akun Admin Baru



Gambar 3. 1. 10 Tampilan Menu Menambahkan Akun Admin Baru

Gambar menampilkan menu tambahkan akun admin dimana admin di minta untuk memasukkan username dan password. Jika inputan valid maka akan menampilkan pesan berhasil. Kemudian program meminta untuk memasukkan pilihan.

11. Menu Menampilkan Daftar Pengunjung



Gambar 3. 1. 11 Tampilan Menu Menampilkan Daftar Pengunjung

Gambar menampilkan menu menampilkan daftar pengguna. Semua pengunjung yang mengunjungi program akan ditampilkan yang terdiri dari username dan rolenya. Kemudian program meminta untuk memasukkan pilihan.

12. Menu Menghapus Akun Pengguna Biasa

```
Daftar Pengguna Biasa:
  username
0    USER
Masukkan username pengunjung yang ingin dihapus: USER
Akun USER berhasil dihapus.

=====
~  MENU
~
~
~
~
~
~
~
~
~
~
=====

=====
| 1. TAMBAHKAN PRODUK |
| 2. TAMPILKAN PRODUK |
| 3. TAMBAHKAN AKUN ADMIN |
| 4. TAMPILKAN DAFTAR PENGUNJUNG |
| 5. HAPUS AKUN PENGUNJUNG |
| 6. UBAH PRODUK |
| 7. HAPUS PRODUK |
| 8. LIHAT KRITIK/SARAN |
| 9. KELUAR |
=====

Pilih menu: █
```

Gambar 3. 1. 12 Tampilan Menu Menghapus Akun Pengguna Biasa

Gambar menampilkan menu menghapus akun pengunjung. Pertama program akan menampilkan semua daftar pengunjung dan program meminta untuk memasukkan username yang ingin dihapus. Jika valid maka akan menampilkan pesan berhasil. Kemudian program meminta untuk memasukkan pilihan.

13. Menu Mengubah Produk

```
+-----+
| Kode Produk | Nama      | Warna   | Stok  | Harga (Rp) |
+-----+-----+-----+-----+
| 1           | pahmina  | merah   | 0     | 20,000    |
| 2           | Segitiga | Ungu    | 20    | 30,000,000|
| 3           | Khimar   | Hitam   | 80    | 10,000,000|
| 4           | Bengo    | merah   | 70    | 50,000,000|
| 5           | Scraf    | Jingga  | 100   | 15,000,000|
+-----+-----+-----+-----+

Masukkan kode produk yang ingin diubah: 2

Data produk saat ini:
Nama: Segitiga, Warna: Ungu, Stok: 20, Harga: 30000000
Nama baru (kosongkan untuk tidak mengubah):
Warna baru (kosongkan untuk tidak mengubah):
Stok baru (kosongkan untuk tidak mengubah): 800
Harga baru (kosongkan untuk tidak mengubah):

Produk 2 berhasil diperbarui.
```

```
=====
~~~MENU~~~~
=====

ADMINI~
=====
```

```
=====
|          |          |          |          |          |
| 1. TAMBAHAKAN PRODUK |          |          |          |          |
| 2. TAMPILKAN PRODUK  |          |          |          |          |
| 3. TAMBAHAKAN AKUN ADMIN |        |          |          |          |
| 4. TAMPILKAN DAFTAR PENGUNJUNG |       |          |          |          |
| 5. HAPUS AKUN PENGUNJUNG |      |          |          |          |
| 6. UBAH PRODUK         |     |          |          |          |
| 7. HAPUS PRODUK        |    |          |          |          |
| 8. LIHAT KRITIK/SARAN  |   |          |          |          |
| 9. KELUAR              |  |          |          |          |
|          |          |          |          |          |
=====
```

```
Pilih menu:
```

Gambar 3. 1. 13 Tampilan Menu Mengubah Produk

Gambar menampilkan menu mengubah produk. Dimulai dengan menampilkan semua produk hijab yang ada. Kemudian program meminta memasukkan kode produk yang ingin diubah. Kemudian program akan menampilkan produk yang ingin dirubah. Selanjutnya program akan meminta untuk memasukkan nama baru, warna baru, stok baru dan harga baru namun jika tidak mau dirubah diantar item tersebut cukup dengan mengosongkan dan tekan enter. Kemudian program meminta untuk memasukkan pilihan.

14. Menu Menghapus Produk

```
+-----+-----+-----+-----+-----+
| Kode Produk | Nama   | Warna | Stok | Harga (Rp) |
+-----+-----+-----+-----+-----+
| 1           | pahmina | merah | 0    | 20,000     |
| 2           | Segitiga | Ungu  | 800  | 30,000,000 |
| 3           | Khimar  | Hitam | 80   | 10,000,000 |
| 4           | Bergo   | merah | 70   | 50,000,000 |
| 5           | Scraf   | Jingga | 100  | 15,000,000 |
+-----+-----+-----+-----+-----+
Masukkan kode produk yang ingin dihapus: 5

Hijab 'Scraf' berhasil dihapus.

=====
~ ~ ~ ~ ~
~ ~ MENU ~ ~ ~ ~ ~
~ ~ ~ ~ ~
~ ~ ADMIN ~ ~ ~ ~ ~
~ ~ ~ ~ ~
=====

=====
| 1. TAMBAHKAN PRODUK |
| 2. TAMPILKAN PRODUK |
| 3. TAMBAHKAN AKUN ADMIN |
| 4. TAMPILKAN DAFTAR PENGUNJUNG |
| 5. HAPUS AKUN PENGUNJUNG |
| 6. UBAH PRODUK |
| 7. HAPUS PRODUK |
| 8. LIHAT KRITIK/SARAN |
| 9. KELUAR |
=====

Pilih menu: |
```

Gambar 3. 1. 14 Tampilan Menu Menghapus Produk

Gambar menampilkan menu menghapus produk. Program akan menampilkan semua produk hijab yang ada. Kemudian program meminta untuk memasukkan kode produk yang ingin berhasil. Jika valid maka akan menampilkan pesan berhasil. Dan produk akan terhapus. Kemudian program meminta untuk memasukkan pilihan.

15. Menu Lihat Kritik/Saran



Gambar 3. 1. 15 Tampilan Menu Lihat Kritik/Saran

Gambar menampilkan menu kritik/saran. Dimana menu ini akan menampilkan semua kritik saran yang sudah dimasukkan oleh pengguna mengenai toko atau produk. Kemudian program meminta untuk memasukkan pilihan.

3.2 Source Code

```
import pandas as pd #Mengimport pustaka pandas untuk mengelola data dalam format dataframe
import csv #Mengimport pustaka csv untuk membaca dan menulis file CSV

from colorama import init, Fore, Style #Mengimport pustaka colorama untuk memberikan warna pada teks di terminal
import os

import platform

from prettytable import PrettyTable #Mengimport Prettytable untuk membuat table pada tampilan menu

#LIST

akun_admin = [['admin', 'admin', 'admin']] #Menyimpan data akun admin

#TUPLE

users = [] #Daftar untuk menyimpan data users
pesanan = [] #Daftar untuk menyimpan data pesanan
kritik_saran = [] #Daftar untuk menyimpan data kritik dan saran

def clear_terminal():

    current_os = platform.system().lower() #Mengecek sistem operasi

    if current_os == "windows":

        os.system('cls') #Untuk Windows

    else:

        os.system('clear') #Untuk Linux/MacOS

clear_terminal() #Memanggil fungsi untuk membersihkan terminal

def muat_data_hijab():

    try:

        with open("dataproduk.csv", mode="r", encoding="utf-8") as file:

            reader = csv.DictReader(file)

            return {row["kode_produk"]: row for row in reader}

    except FileNotFoundError:

        print("File 'dataproduk.csv' tidak ditemukan. Mulai dengan data kosong.")

        return {}

data_hijab = muat_data_hijab() #Menyimpan data produk hijab

def simpan_data_hijab():

    if not data_hijab: #Jika data_hijab kosong

        with open('dataproduk.csv', mode='w', newline='') as file: #Membuka file CSV untuk ditulis

            writer = csv.writer(file)

            writer.writerow(['kode_produk', 'nama', 'warna', 'stok', 'harga']) # Menulis header CSV

    else:

        df = pd.DataFrame([[]
```

```

        'kode_produk': code,

        'nama': item['nama'],

        'warna': item['warna'],

        'stok': item['stok'],

        'harga': item['harga']

    } for code, item in data_hijab.items()] #Mengubah dictionary data_hijab menjadi DataFrame

    df.to_csv('dataproduk.csv', index=False) #Menyimpan DataFrame ke dalam file CSV

def simpan_data_pengguna():

    if users:

        df = pd.DataFrame(users)

        df.to_csv('data_pengguna.csv', index=False)

def muat_data_pengguna():

    try:

        df = pd.read_csv('data_pengguna.csv')

        for _, row in df.iterrows():

            users.append({

                'username': row['username'],

                'password': row['password'],

                'role': row['role']

            })

    except FileNotFoundError:

        pass

def registrasi_user(username, password):

    if not username.strip():

        raise ValueError("Username tidak boleh kosong")

    if not password.strip():

        raise ValueError("Password tidak boleh kosong")

    for user in users:

        if user["username"] == username:

            raise ValueError("Username telah terdaftar!")

    users.append({

        "username": username,

        "password": password,

        "role": "pengguna biasa"

    })

    simpan_data_pengguna()

    print(f"Registrasi berhasil, {username} telah ditambahkan sebagai pengguna biasa.")

```

```

def login():
    try:
        # Membaca data pengguna dari file CSV
        df = pd.read_csv('data_pengguna.csv')
    except FileNotFoundError:
        raise ValueError("Data pengguna belum ada. Silakan registrasi terlebih dahulu.")

    # Meminta username dan password dari pengguna
    username = input("Username: ")
    password = input("Password: ")

    # Memeriksa data di dalam file CSV
    for _, row in df.iterrows():
        if row["username"] == username and row["password"] == password:
            return row["role"] # Mengembalikan peran pengguna jika cocok

    # Memeriksa kredensial admin
    for akun in akun_admin:
        if username == akun[0] and password == akun[1]:
            return akun[2]

    # Jika tidak cocok, login gagal
    raise ValueError("Login gagal!! Username atau password salah.")

def tampilkan_daftar_pengunjung():
    # Muat data dari file CSV
    pengguna_csv = []

    try:
        df = pd.read_csv('data_pengguna.csv')

        for _, row in df.iterrows():
            pengguna_csv.append({
                'username': row['username'],
                'role': row['role']
            })
    except FileNotFoundError:
        print("File 'data_pengguna.csv' tidak ditemukan. Memuat data hanya dari memori.")

    # Gabungkan data dari CSV dan dari variabel users
    semua_pengguna = {user['username']: user for user in pengguna_csv} # Hindari duplikasi dengan dictionary

    for user in users:
        if user['username'] not in semua_pengguna: # Tambahkan hanya jika belum ada
            semua_pengguna[user['username']] = user

```

```

# Tampilkan data pengguna

if not semua_pengguna:

    print("Tidak ada pengguna yang terdaftar.")

else:

    print("Daftar Pengguna:")

    for username, data in semua_pengguna.items():

        print(f"Username: {username}, Role: {data['role']}")

def tampilkan_produk():

    if not data_hijab: # Jika tidak ada produk hijab yang tersedia

        print("Tidak ada produk hijab yang tersedia.")

        return

    table = PrettyTable() # Membuat tabel dengan header

    table.field_names = ["Kode Produk", "Nama", "Warna", "Stok", "Harga (Rp)"]

    for kode_produk, item in data_hijab.items():

        try:

            # Pastikan harga dan stok adalah integer

            stok = int(item["stok"])

            harga = int(item["harga"])

            table.add_row([

                kode_produk,

                item["nama"],

                item["warna"],

                stok,

                f"{harga:,}" # Format harga dengan pemisah ribuan

            ])

        except ValueError as e:

            print(f>Data tidak valid untuk produk {kode_produk}: {e}")

    print(table)

def tambah_produk():

    try:

        nama = input("Nama: ").strip() #Meminta input nama produk dan validasi agar hanya berisi huruf

        if not nama or not nama.isalpha():

            raise ValueError("Nama harus berupa huruf dan tidak boleh kosong!")

        warna = input("Warna: ").strip() #Meminta input warna produk dan validasi agar hanya berisi huruf

        if not warna or not warna.isalpha():

            raise ValueError("Warna harus berupa huruf dan tidak boleh kosong!")

```

```

        stok_input = input("Stok: ").strip() #Meminta input stok produk dan validasi agar berupa angka positif
        if not stok_input.isdigit() or int(stok_input) <= 0:
            raise ValueError("Stok harus berupa angka positif dan tidak boleh kosong!")

        stok = int(stok_input)

        harga_input = input("Harga: ").strip() #Meminta input harga produk dan validasi agar berupa angka positif
        if not harga_input.isdigit() or int(harga_input) <= 0:
            raise ValueError("Harga harus berupa angka positif dan tidak boleh kosong!")

        harga = int(harga_input)

        kode_produk = str(len(data_hijab) + 1) #Membuat kode produk berdasarkan jumlah produk yang ada

        data_hijab[kode_produk] = { #Menambahkan produk baru ke dalam data_hijab

            "nama": nama,

            "warna": warna,

            "stok": stok,

            "harga": harga

        }

        simpan_data_hijab() #Menyimpan data produk hijab

        print(f"\nHijab {nama} berhasil ditambahkan.") #Menampilkan pesan sukses

    except ValueError as e:

        print(f"Input tidak valid: {e}") #Menangkap dan menampilkan pesan error untuk input tidak valid

def ubah_produk():

    try:

        if not data_hijab: # Mengecek apakah tidak ada produk dalam data_hijab

            print("Tidak ada produk hijab yang tersedia.")

            return False

        tampilkan_produk() # Menampilkan semua produk hijab yang tersedia

        kode_produk = input("Masukkan kode produk yang ingin diubah: ").strip()

        if kode_produk not in data_hijab: # Mengecek apakah kode produk valid

            print("Kode produk tidak ditemukan!")

            return False

        # Menampilkan data produk yang akan diubah

        produk_lama = data_hijab[kode_produk]

        print("\nData produk saat ini:")

        print(f"Nama: {produk_lama['nama']}, Warna: {produk_lama['warna']}, Stok: {produk_lama['stok']}, Harga: {produk_lama['harga']}")

```

```

# Meminta input data baru

nama_baru = input("Nama baru (kosongkan untuk tidak mengubah): ").strip()

warna_baru = input("Warna baru (kosongkan untuk tidak mengubah): ").strip()

stok_baru_input = input("Stok baru (kosongkan untuk tidak mengubah): ").strip()

harga_baru_input = input("Harga baru (kosongkan untuk tidak mengubah): ").strip()

# Gunakan data lama jika input kosong

nama_baru = nama_baru if nama_baru else produk_lama['nama']

warna_baru = warna_baru if warna_baru else produk_lama['warna']

stok_baru = int(stok_baru_input) if stok_baru_input.isdigit() else int(produk_lama['stok'])

harga_baru = int(harga_baru_input) if harga_baru_input.isdigit() else int(produk_lama['harga'])

# Validasi input

if not nama_baru.isalpha():

    raise ValueError("Nama harus berupa huruf!")

if not warna_baru.isalpha():

    raise ValueError("Warna harus berupa huruf!")

if stok_baru <= 0:

    raise ValueError("Stok harus berupa angka positif!")

if harga_baru <= 0:

    raise ValueError("Harga harus berupa angka positif!")

# Perbarui data produk

data_hijab[kode_produk] = {

    "nama": nama_baru,

    "warna": warna_baru,

    "stok": stok_baru,

    "harga": harga_baru

}

# Simpan perubahan ke file CSV

simpan_data_hijab()

print(f"\nProduk {kode_produk} berhasil diperbarui.")

return True

except ValueError as e: # Menangani error jika input tidak valid

    print(f"Input tidak valid: {e}")

    return False

def hapus_produk():

    try:

        # Mengecek apakah ada produk

        if not data_hijab:

```



```

        print("Tidak ada produk hijab yang tersedia.")

        return

    # Menampilkan daftar produk
    tampilkan_produk()

    # Meminta input kode produk untuk dihapus
    kode_produk = input("Masukkan kode produk yang ingin dihapus: ").strip()

    # Mengecek apakah produk ada
    if kode_produk in data_hijab:
        nama_produk = data_hijab[kode_produk]["nama"]

        # Hapus produk dari dictionary
        del data_hijab[kode_produk]

        # Simpan data yang diperbarui ke CSV
        simpan_data_hijab()

        # Perbarui data_hijab dari CSV
        muat_data_hijab()

        print(f"\nHijab '{nama_produk}' berhasil dihapus.")
    else:
        print("\nKode produk tidak valid atau tidak ditemukan!")

except Exception as e:
    print(f"Terjadi kesalahan: {e}")

def muat_data_kritik_saran():
    try:
        df = pd.read_csv('kritik_saran.csv') #Mencoba untuk membaca file CSV 'kritik_saran.csv'

        for index, row in df.iterrows(): #Uterasi setiap baris dalam DataFrame
            kritik_saran.append({
                'username': row['username'], #Menambahkan data username
                'kritik': row['kritik'] #Menambahkan data kririk/saran
            })
    except FileNotFoundError: #Jika file tidak ditemukan
        pass #Tidak melakukan apa-apa, cukup lewatkan

def simpan_data_kritik_saran():
    if kritik_saran: #Mengecek apakah kritik atau saran yang tersimpan dalam list
        df = pd.DataFrame(kritik_saran) #Mengubah list kritik_saran menjadi DataFrame
        df.to_csv('kritik_saran.csv', index=False) #Menyimpan DataFrame ke file CSV
    else:
        print("Tidak ada kritik atau saran untuk disimpan.") #Menampilkan pesan jika tidak ada kritik/saran

```

```

def masukan_kritik(username):

    kritik = input("Masukkan kritik atau saran Anda: ") #Meminta pengguna untuk menginputkritik atau saran

    kritik_saran.append({ #Menambahkan kritik atau saran ke dalam list kritik_saran

        'username': username, #Menyimpan username pengguna yang memberikan kritik

        'kritik': kritik #Menyimpan kritik atau saran dari pengguna

    })

    simpan_data_kritik_saran() #Memanggil fungsi untuk menyimpan kritik dan saran ke file

    print("Terima kasih atas kritik dan sarannya!") #Menampilkan pesan konfirmasi

    menu_pengguna_biasa() #Kembali ke menu pengguna biasa

def tampilkan_kritik_saran():

    if not kritik_saran: #Mengecek apakah list kritik_saran kosong

        print("Belum ada kritik atau saran yang masuk.") #Menampilkan pesan jika tidak ada kritik/saran

    else:

        print("Daftar Kritik dan Saran:") #Menampilkan header daftar kritik dan saran

        for entry in kritik_saran: #Iterasi setiap kritik yang ada dalam list kritik_saran

            print(f"Pengguna: {entry['username']}, Kritik/Saran: {entry['kritik']}") #Menampilkan username dan kritik/saran

def simpan_data_pesanan():

    df = pd.DataFrame(pesanan) #Mengubah list pesanan menjadi DataFrame

    df.to_csv('pesanan.csv', index=False) #Menyimpan DataFrame ke file CSV 'pesanan.csv'

# Misalkan username sudah didefinisikan pada proses login sebelumnya

def buat_pesanan():

    try:

        # Memuat data produk dari file CSV ke dalam dictionary

        df = pd.read_csv('dataproduk.csv')

        data_hijab = {

            row['kode_produk']: {

                'nama': row['nama'],

                'warna': row['warna'],

                'stok': int(row['stok']),

                'harga': int(row['harga'])

            }

            for _, row in df.iterrows()

        }

        if not data_hijab: # Mengecek apakah data_hijab kosong

            print("Tidak ada produk hijab yang tersedia.")

            menu_pengguna_biasa()

            return

        # Menampilkan produk hijab yang tersedia

        tampilkan_produk()

```

```

# Meminta input kode produk

kode_produk = int(input("Masukkan kode produk yang ingin Anda pesan: ").strip())

if kode_produk in data_hijab: # Memeriksa apakah kode produk valid

    try:

        # Meminta input jumlah produk yang ingin dipesan

        jumlah = int(input("Masukkan jumlah yang ingin dipesan: "))

        if jumlah > data_hijab[kode_produk]['stok']: # Memeriksa stok

            print("Jumlah stok tidak mencukupi!")

            menu_pengguna_biasa()

            return

        # Mengurangi stok produk

        data_hijab[kode_produk]['stok'] -= jumlah

        # Menyimpan data produk kembali ke CSV

        updated_df = pd.DataFrame([

            'kode_produk': kode,

            'nama': item['nama'],

            'warna': item['warna'],

            'stok': item['stok'],

            'harga': item['harga']

        ] for kode, item in data_hijab.items())

        updated_df.to_csv('dataproduk.csv', index=False)

        # Meminta input username di sini

        username = input("Masukkan username Anda: ").strip()

        # Menambahkan pesanan ke dalam daftar pesanan

        pesanan.append({

            'username': username, # Menggunakan username yang baru dimasukkan

            'kode_produk': kode_produk,

            'nama_produk': data_hijab[kode_produk]['nama'],

            'jumlah': jumlah,

            'harga_total': jumlah * data_hijab[kode_produk]['harga']

        })

        simpan_data_pesanan()

        print("Pesanan berhasil dibuat.")

        menu_pengguna_biasa()

    except ValueError:

        print("Jumlah harus berupa angka!")

        menu_pengguna_biasa()

else:

```

```

        print("Kode produk tidak valid!")

        menu_pengguna_biasa()

    except FileNotFoundError:

        print("File produk tidak ditemukan. Silakan tambahkan produk terlebih dahulu.")

        menu_pengguna_biasa()

def tampilkan_pesanan():

    if not pesanan: #Memeriksa apakah list pesanan kosong

        print("Belum ada pesanan yang masuk.") #Menampilkan pesan jika pesanan kosong

    else:

        print("Daftar Pesanan:") #Jika ada pesana, tampilkan header "Daftar Pesanan"

        for order in pesanan:

            print(f"Pengguna: {order['username']}, Produk: {order['nama_produk']}, Jumlah: {order['jumlah']}, Total Harga: Rp{order['harga_total']}") #Menampilkan informasi pesanan

def tambah_akun_admin():

    username = input("Username baru untuk admin: ") #Meminta input username

    password = input("Password untuk admin: ") #Meminta input password

    akun_admin.append([username, password, "admin"])

    print(f"Akun admin {username} berhasil ditambahkan.") #Menampilkan pesan konfirmasi

def hapus_akun_pengunjung():

    try:

        # Membaca file CSV ke dalam DataFrame

        df = pd.read_csv('data_pengguna.csv')

        # Menampilkan daftar pengguna biasa untuk memudahkan

        pengguna_biasa = df[df['role'] == 'pengguna biasa']

        if pengguna_biasa.empty:

            print("Tidak ada akun pengunjung biasa yang terdaftar.")

            return

        print("\nDaftar Pengguna Biasa:")

        print(pengguna_biasa[['username']])

        # Meminta input username dari pengguna yang ingin dihapus

        username = input("Masukkan username pengunjung yang ingin dihapus: ").strip()

        # Memeriksa apakah username ada dan merupakan pengguna biasa

        if not ((df['username'] == username) & (df['role'] == 'pengguna biasa')).any():

            print("Akun tidak ditemukan atau bukan akun pengunjung biasa.")

            return

        # Menghapus baris dari DataFrame

```

[illegible]

[illegible]

```

    """)

    pilihan_admin = input("Pilih menu: ") #Meminta input dari admin untuk memilih menu

    if pilihan_admin == '1': #Jika pilihan adalah 1

        clear_terminal()

        tambah_produk() #Admin akan menambahkan produk

        menu_admin() #Kembali ke menu admin

    elif pilihan_admin == '2': #Jika pilihan adalah 2

        clear_terminal()

        tampilkan_produk() #Akan menampilkan daftar produk hijab

        menu_admin() #Kembali ke menu admin

    elif pilihan_admin == '3': #Jika pilihan adalah 3

        clear_terminal()

        tambah_akun_admin() #Akan menambahkan akun admin

        menu_admin() #Kembali ke menu admin

    elif pilihan_admin == '4': #Jika pilihan adalah 4

        clear_terminal()

        tampilkan_daftar_pengunjung() #Akan menampilkan daftar pengunjung

        menu_admin() #Kembali ke menu admin

    elif pilihan_admin == '5': #Jika pilihan adalah 5

        clear_terminal()

        hapus_akun_pengunjung() #Admin akan menghapus akun pengunjung

        menu_admin() #Kembali ke menu admin

    elif pilihan_admin == '6': #Jika pilihan adalah 6

        clear_terminal()

        ubah_produk() #Admin akan mengubah produk hijab

        menu_admin() #Kembali ke menu admin

    elif pilihan_admin == '7': #Jika pilihan adalah 7

        clear_terminal()

        hapus_produk() #Admin akan menghapus produk hijab

        menu_admin() #Kembali ke menu admin

    elif pilihan_admin == '8': #Jika pilihan adalah 8

        clear_terminal()

        tampilkan_kritik_saran() #Akan menampilkan daftar kritik/saran

        menu_admin() #Kembali ke menu admin

    elif pilihan_admin == '9': #Jika pilihan adalah 9

        clear_terminal()

        print("Terimakasih") #Admin akan keluar dari menu admin dan menampilkan pesan konfirmasi

    else:

        print("Pilihan tidak valid!!") #Jika pilihan tidak valid dan menampilkan pesan kesalahan

        menu_admin() #Kembali ke menu admin

    muat_data_hijab() #Memuat data produk hijab dari file atau database

    while True:

        try:

```

[illegible]

BAB IV

PENUTUP

4.1 Kesimpulan

Program ini dirancang untuk memudahkan proses pembelian toko hijab , memberikan kemudahan bagi pengguna dalam memilih, membeli, dan mengelola transaksi toko hijab. Dengan antarmuka yang ramah pengguna dan fungsi yang efisien, program ini diharapkan dapat meningkatkan kualitas layanan dan mengoptimalkan proses administrasi pada bisnis toko hijab. Program ini dibangun menggunakan bahasa pemrograman Python yang memungkinkan pengelolaan data yang terstruktur dan mudah dikembangkan lebih lanjut.

4.2 Saran

Kami menyadari bahwa program ini masih dapat ditingkatkan di berbagai aspek, baik dari segi fungsionalitas maupun penyajian data. Masukan berupa kritik dan saran yang membangun dari para pembaca akan sangat kami hargai untuk penyempurnaan program ini di masa mendatang. Kami berharap program ini bermanfaat dan dapat digunakan sebagai solusi dalam pengelolaan toko hijab secara efektif.



DAFTAR PUSTAKA

- Inpows. (2020, Oktober 30). *Membuat Tabel Menggunakan PrettyTable di Python*. Retrieved November 2024, 01, from inpows.com:
<https://www.inpows.com/python/membuat-tabel-menggunakan-prettytable-di-python/>
- Muhardian, A. (2019, Desember 25). *Belajar Membuat Aplikasi CRUDS dengan Python dan File CSV*. Retrieved November 07, 2024, from petanikode.com: <https://www.petanikode.com/python-csv-crud/>
- Widiarto, S. (2024, Juli 17). *CRUD: Pengertian, Fungsi, Contoh, dan Cara Membuatnya*. Retrieved Oktober 27, 2024, from rumahweb.com: <https://www.rumahweb.com/journal/crud-adalah/>



LAMPIRAN



Gambar 1 Cover Kartu Konsul

Tanggal Konsultasi : Kamis, 31 okt 2024	
Uraian / Pembahasan : flowchart : ditambah off page Menu admin : . bisa membuat akun admin baru . bisa menghapus akun user. Menu pengguna : . setiap item ditambahkan harga total.	
Asisten Lab  Nama: Celio Arga .R	Ketua Kelompok  Nama: Syafiq Hafizh F

Gambar 2 Kartu Konsultasi Pertama

Tanggal Konsultasi : Rabu, 13 November 2024	
Uraian / Pembahasan : <ul style="list-style-type: none"> - Error Handling - flowchart: <ul style="list-style-type: none"> • start diperbaiki • Nama page - Sistem Clear terminal 	
Asisten Lab  Nama: Celio Arga R.	Ketua Kelompok  Nama: Syafiq Hafizh F.

Gambar 3 Kartu Konsultasi Kedua