

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



**SISTEM MANAJEMEN LAYANAN PEMESANAN DAN  
PENGIRIMAN**

**Oleh:**

**Kelompok 1**

<b>AYU AZZHAHRAH ALWI</b>	<b>2409106024</b>
<b>MUHAMMAD HAYKAL</b>	<b>2409106005</b>
<b>MAKHMUD</b>	
<b>GHESYA RHEGYTA AL</b>	<b>2409106023</b>
<b>RACHMAN</b>	

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

## **KATA PENGANTAR**

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga kami dapat menyelesaikan laporan penilaian akhir mata kuliah Algoritma dan Pemrograman Dasar ini tepat pada waktunya. Laporan ini disusun sebagai bagian dari proyek akhir praktikum dan bertujuan untuk mendokumentasikan pemahaman dan keterampilan yang telah kami pelajari dalam menerapkan konsep-konsep dasar algoritma dan pemrograman.

Program yang kami buat merupakan sebuah program manajemen pemesanan layanan yang dikembangkan menggunakan bahasa pemrograman python. Program manajemen pemesanan layanan ini menggunakan teknologi geolocation untuk menentukan lokasi penjemputan dan lokasi tujuan. Program ini juga menghitung total harga yang harus dibayar berdasarkan layanan, berat barang, dan jarak yang ditempuh untuk mengantarkan. Program pengiriman barang ini menyediakan beberapa jasa layanan pengiriman seperti mobil, motor, dan kurir.

Dalam proses pembuatan program kami terkendala ketika mengintegrasikan teknologi geolocation karena memerlukan pemahaman mendalam tentang penggunaan API dan pengolahan data lokasi. Tetapi karena adanya bantuan dari berbagai pihak, kami berhasil mengatasi kendala-kendala tersebut. Kami juga ingin mengucapkan terima kasih yang sebesar-besarnya kepada dosen pengampu mata kuliah serta asisten dosen yang telah memberikan ilmu dan masukan selama perkuliahan.

Demikianlah laporan akhir ini kami buat. Kami menyadari bahwa masih banyak kekurangan dalam program yang telah kami buat, namun kami berharap laporan ini dapat memberikan gambaran tentang kemampuan kami. Kami juga berharap dapat mengembangkan kemampuan kami dalam bidang pemrograman agar dapat memberikan manfaat pada dunia teknologi kedepannya.

Samarinda, 06 November 2024

Kelompok 1

## TAKARIR

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

<i>Login</i>	Masuk
<i>Register</i>	Daftar
<i>Input</i>	Memasukkan
<i>Admin menu</i>	Menu admin
<i>User menu</i>	Menu pengguna
<i>Flowchart</i>	Diagram alir
<i>Main menu</i>	Menu utama
<i>Error</i>	Kesalahan
<i>Dashboard</i>	Dasbor
<i>Username</i>	Nama pengguna
<i>Password</i>	Kata sandi
<i>Role</i>	Peran
<i>Manage</i>	Kelola
<i>History</i>	Riwayat
<i>Start</i>	Mulai
<i>End</i>	Selesai
<i>Valid</i>	Benar
<i>Source Code</i>	Sumber kode
<i>Real-time</i>	Waktu nyata
<i>Exit</i>	Keluar
<i>Logout</i>	Keluar

## DAFTAR ISI

KATA PENGANTAR .....	2
TAKARIR.....	3
DAFTAR ISI.....	4
DAFTAR GAMBAR .....	6
BAB I PENDAHULUAN .....	7
1.1 Latar Belakang .....	7
1.2 Kebutuhan Fungsional .....	7
1.3 Rumusan Masalah.....	8
1.4 Batasan Masalah .....	8
1.5 Tujuan .....	8
BAB II PERANCANGAN .....	10
2.1 Analisis Program.....	10
2.2 Flowchart .....	11
2.3 Konsep/Materi Praktikum yang dipakai .....	21
BAB III HASIL DAN PEMBAHASAN .....	24
3.1 Tampilan Program .....	24
1. Tampilan Menu.....	24
2. Tampilan menu user.....	24
3. Tampilan menu layanan .....	24
4. Tampilan layanan motor .....	25
5. Tampilan layanan mobil.....	25
6. Tampilan layanan kurir .....	26
7. History pesanan.....	26
8. Menu admin .....	27
9. Konfirmasi pesanan.....	27
10. Manage user .....	28
11. Manage Layanan .....	28
12. Manage pesanan.....	28

3.2 Source Code .....	29
BAB IV PENUTUP .....	46
4.1 Kesimpulan .....	46
4.2 Saran .....	46
DAFTAR PUSTAKA .....	47
LAMPIRAN.....	48

## DAFTAR GAMBAR

Gambar 2.1 Flowchart Manajemen Layanan Main Flow .....	11
Gambar 2.2 Flowchart Login .....	13
Gambar 2.3 Flowchart Register .....	14
Gambar 2.4 Flowchart Admin Menu .....	15
Gambar 2.5 Flowchart Manage User .....	16
Gambar 2.6 Flowchart Manage Layanan .....	17
Gambar 2.7 Manage Pesanan .....	18
Gambar 2.8 Flowchart Menu User .....	19
Gambar 2.9 Flowchart Pemesanan .....	20
Gambar 3.1 Tampilan Menu .....	24
Gambar 3.2 Menu User .....	24
Gambar 3.3 Menu Layanan .....	25
Gambar 3.4 Layanan Motor .....	25
Gambar 3.5 Layanan Mobil .....	25
Gambar 3.6 Layanan Kurir .....	26
Gambar 3.7 History Pesanan .....	26
Gambar 3.8 Menu Admin .....	27
Gambar 3.9 Konfirmasi Pesanan .....	27
Gambar 3.10 Manage User .....	28
Gambar 3.11 Manage Layanan .....	28
Gambar 3.12 Manage Pesanan .....	29
Gambar 1 Lampiran Kartu Konsul .....	48
Gambar 2 Lampiran Kartu Konsul .....	49

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pada era globalisasi saat ini, perkembangan teknologi sangatlah pesat terutama dalam bidang jasa pengiriman. Perkembangan teknologi tersebut memberi dampak kepada masyarakat berupa meningkatnya kebutuhan layanan jasa pengiriman yang cepat dan mudah. Tetapi tidak dapat dipungkiri bahwa dalam praktiknya teknologi yang diharapkan dapat mempermudah malah menjadi mempersulit pengguna ketika ingin menggunakan teknologi tersebut. Tidak hanya mempersulit pengguna biasa, bahkan terkadang teknologi tersebut sulit digunakan oleh admin yang bertugas untuk mengelola layanan dan pesanan.

Oleh karena itu, kami membuat sistem jasa pengiriman barang ini, yang dapat mempermudah admin dalam mengelola data berupa data layanan serta data pengguna teknologi tersebut. Admin dapat mengubah, menambahkan, ataupun menghapus layanan yang tersedia, mengelola data pengguna yang telah terdaftar, serta memproses dan mengonfirmasi pesanan. Sistem pemesanan layanan ini dilengkapi dengan geolocation yang berfungsi untuk menghitung biaya yang harus dibayar berdasarkan jarak antara titik jemput dan titik antar. Dengan adanya teknologi jasa pengiriman ini diharapkan dapat membantu pengguna dan admin dalam melakukan proses pemesanan dan pengiriman barang.

### **1.2 Kebutuhan Fungsional**

1. Login
2. Register
3. Admin menu
4. User menu

5. Main menu

6. Penanganan eror

7. Validasi input

### **1.3 Rumusan Masalah**

Berikut kami sertakan rumusan masalah yang terdapat pada laporan ini, rumusan masalah dibuat dengan tujuan untuk membantu kami dalam merumuskan permasalahan yang akan diselesaikan.

- 1) Bagaimana cara mengelola data layanan, pengguna, dan pesanan?
- 2) Bagaimana cara membangun sistem pemesanan layanan yang mudah dan cepat?
- 3) Bagaimana cara membangun sistem yang dapat mencatat riwayat pesanan dan status transaksi?

### **1.4 Batasan Masalah**

Berikut kami sertakan batasan masalah yang ada pada laporan ini, batasan masalah dibuat agar penyelesaian masalah yang dilakukan tidak terlalu meluas.

- 1) Sistem pemesanan hanya menangani manajemen data layanan, pengguna, dan pesanan berupa tambah, ubah, dan hapus.
- 2) Sistem pemesanan akan dirancang agar mudah digunakan oleh pengguna.
- 3) Sistem pemesanan akan menyimpan data setiap pesanan dan menampilkan status transaksi untuk memudahkan pengguna dalam memantau pesanan.

### **1.5 Tujuan**

Adapun tujuan mengapa sistem ini kami buat adalah sebagai berikut.

- 1) Mengelola data layanan, pengguna, dan pesanan secara cepat dan akurat.
- 2) Membangun sistem pemesanan yang mudah dan cepat.



- 3) Menyediakan sistem pencatatan riwayat pesanan dan status transaksi.

## **BAB II**

### **PERANCANGAN**

#### **2.1 Analisis Program**

Program yang kami buat adalah program pemesanan layanan yang mempunyai fungsi untuk mencatat, mengelola, serta mengelola data layanan, pengguna, dan pesanan. Program yang kami buat memberikan kontrol penuh kepada admin untuk mengelola layanan yang disediakan, pengguna yang terdaftar, dan pesanan yang dibuat oleh pengguna. Tidak hanya itu, program yang kami buat juga memberikan kemudahan bagi pengguna untuk melakukan proses pemesanan. Berikut adalah alur program yang terjadi pada program manajemen layanan yang kami buat.

##### **1) Dashboard admin**

1. Admin melakukan login untuk masuk kedalam dashboard admin.
2. Admin dapat mengubah, menambahkan dan menghapus data layanan tersedia ketika sudah berhasil login.
3. Admin dapat mengelola data pengguna yang sudah tersimpan kedalam sistem.
4. Admin dapat melihat serta mengelola data pesanan yang telah dibuat oleh pengguna yang terdaftar.

##### **2) Proses pemesanan oleh pengguna**

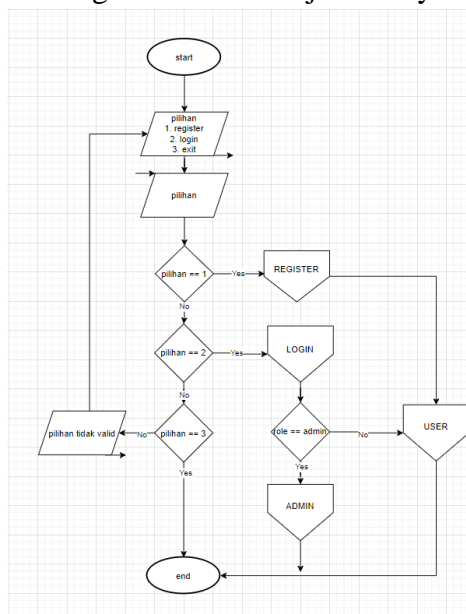
1. Registrasi bagi pengguna yang belum terdaftar pada sistem.
2. Login pengguna menggunakan username dan password yang sudah terdaftar pada sistem.
3. Pengguna memilih layanan.
4. Estimasi pembayaran.
5. Konfirmasi oleh pengguna.

##### **3) Pencatatan riwayat dan status transaksi**

1. Ketika pengguna melakukan konfirmasi pesanan maka sistem akan mencatat pesanan kedalam riwayat pengguna.
2. Pengguna dapat melihat status pesanan dalam riwayat mereka, yang akan diperbarui oleh admin sesuai dengan perkembangan transaksi.
3. Sistem akan memberikan notifikasi kepada pengguna ketika pembaruan status pesanan terjadi.

## 2.2 Flowchart

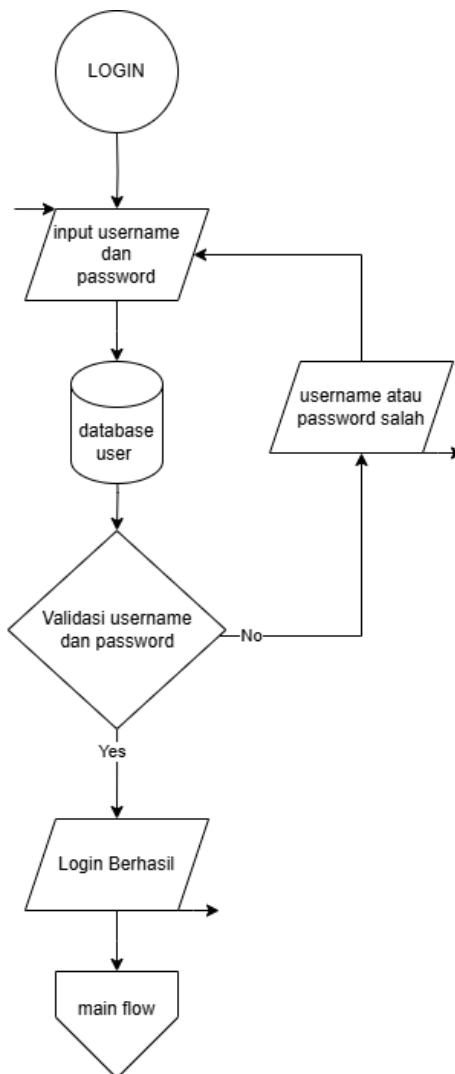
Bagan alir (flowchart) adalah teknik analitis bergambar yang digunakan untuk menjelaskan beberapa aspek dari sistem informasi secara jelas, ringkas, dan logis began alir mencatat cara proses bisnis dilakukan dan cara dokumen mengalir melalui organisasi (Tuasamu et al., 2023). Menurut sitorus (2015:14) pada jurnal yang ditulis oleh (Charis Noijsa et al., 2023), flowchart merupakan urutan logika dari suatu prosedur pemecahan masalah, sehingga flowchart merupakan langkah-langkah penyelesaian masalah yang dituliskan dalam simbol-simbol tertentu. Berikut adalah flowchart yang kami buat dalam proses membangun sistem manajemen layanan.



*Gambar 2.1 Flowchart Manajemen Layanan Main Flow*

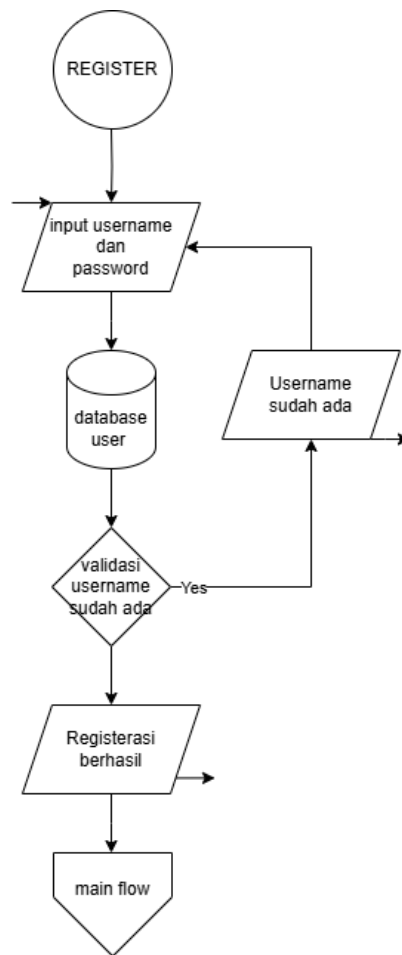
Gambar diatas merupakan flowchart dari sistem pemesanan layanan. Berdasarkan gambar tersebut, alur dari sistem pemesanan layanan tersebut adalah sistem menampilkan sebuah main menu berupa sebuah pilihan yang digunakan oleh pengguna

untuk melakukan register (pendaftaran akun) atau login untuk mengakses sistem pemesanan layanan. Jika pengguna memilih register maka sistem akan meminta pengguna untuk mendaftarkan username dan password agar bisa melakukan proses login. Jika pengguna sudah memiliki username dan password maka pengguna bisa memilih login, kemudian sistem akan memvalidasi role dari pengguna apakah admin atau user. Jika pengguna memiliki role admin maka sistem akan membawa pengguna tersebut ke halaman admin dan sebaliknya jika role pengguna tersebut user maka sistem akan membawa pengguna ke halaman user. Jika pilihan pengguna pada inputan tidak valid maka sistem akan membawa kembali pengguna ke main menu. Jika pengguna memilih exit maka sistem akan mengakhiri program tersebut.



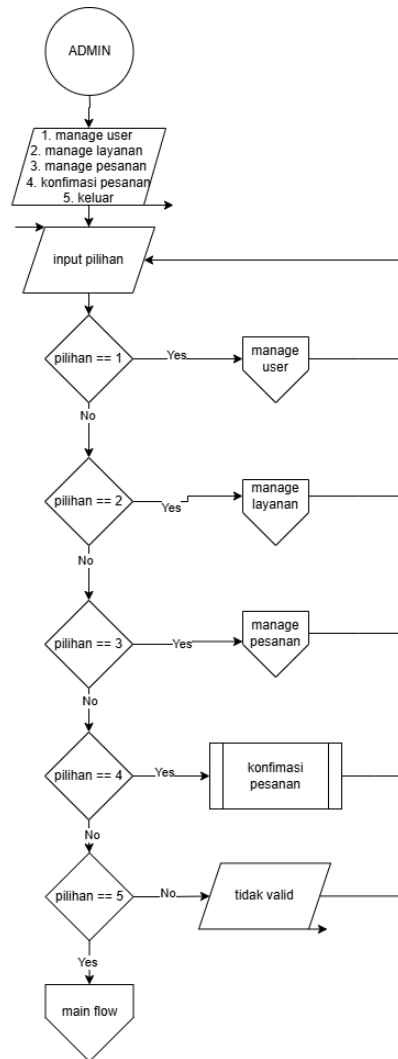
*Gambar 2.2 Flowchart Login*

Gambar 2.2 adalah gambar flowchart login, pada bagian login pengguna diminta untuk memasukkan username dan password yang sudah terdaftar kemudian sistem akan melakukan validasi terhadap data yang tersimpan pada database. Jika username dan password benar maka sistem akan menampilkan login berhasil dan berpindah ke halaman main flow dan sebaliknya jika username atau password salah maka sistem akan meminta pengguna untuk memasukkan username dan passwordnya kembali.



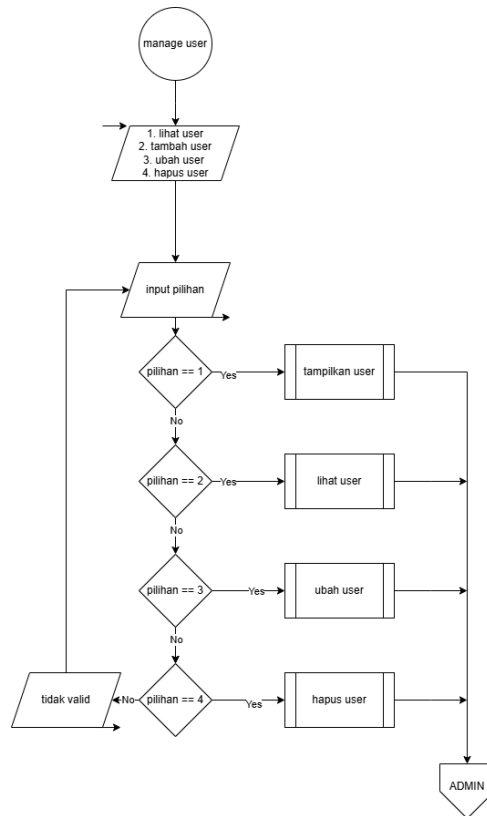
*Gambar 2.3 Flowchart Register*

Gambar 2.3 merupakan flowchart dari halaman register, pada halaman tersebut pengguna akan diminta untuk memasukkan username dan password kemudian data tersebut akan disimpan kedalam database yang selanjutnya akan divalidasi. Apabila berhasil maka sistem akan menampilkan pesan registrasi berhasil, tetapi jika hasil validasi berupa username dan password sudah ada maka pengguna akan diminta untuk memasukkan data username dan password lagi.



*Gambar 2.4 Flowchart Admin Menu*

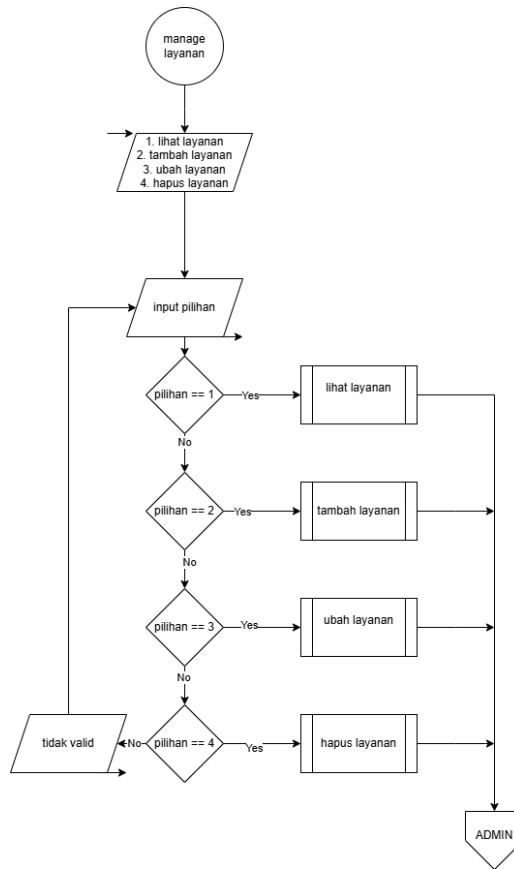
Gambar 2.4 merupakan gambar menu admin yang tampil ketika admin login kedalam sistem. Menu tersebut akan meminta admin untuk memasukkan pilihan dimana jika admin menginputkan 1 maka sistem akan menampilkan menu manage user, jika admin menginputkan 2 maka sistem akan menampilkan menu manage layanan, jika admin menginputkan 3 maka sistem akan menampilkan menu manage pesanan, jika admin menginputkan 4 maka sistem akan menampilkan menu konfirmasi pesanan, dan jika admin menginputkan 5 maka sistem akan berakhir dan kembali ke halaman main flow tetapi jika admin menginputkan angka-angka tersebut maka sistem akan menganggap hal tersebut tidak valid dan akan kembali ke menu input pilihan.



*Gambar 2.5 Flowchart Manage User*

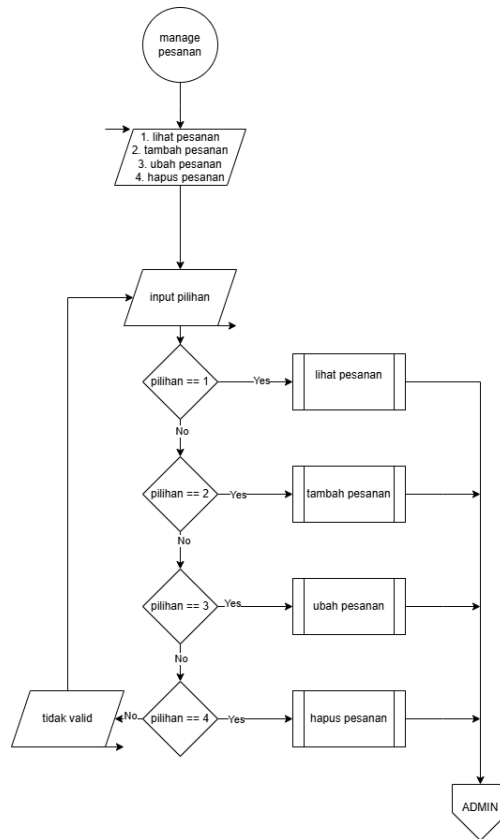
Gambar 2.5 merupakan flowchart yang menggambarkan halaman manage user yang di kelola oleh admin. Halaman tersebut akan menampilkan beberapa pilihan berupa lihat user, tambah user, ubah user, dan hapus user. Setelah itu admin akan diminta untuk menginputkan pilihan dimana jika admin menginputkan 1 maka sistem akan menampilkan data user yang terdaftar, jika admin menginputkan 2 maka sistem akan menampilkan data user yang bisa ditambahkan, jika admin menginputkan 3 maka sistem akan menampilkan halaman untuk mengubah data user, dan jika admin menginputkan 4 maka sistem akan menampilkan halaman yang digunakan untuk menghapus data user. Setiap kali admin selesai melakukan proses inputan tersebut maka sistem akan kembali ke halaman admin, tetapi jika admin menginputkan angka selain angka-angka diatas maka sistem akan menganggap angka tersebut tidak valid dan meminta admin untuk kembali menginputkan angka yang valid.





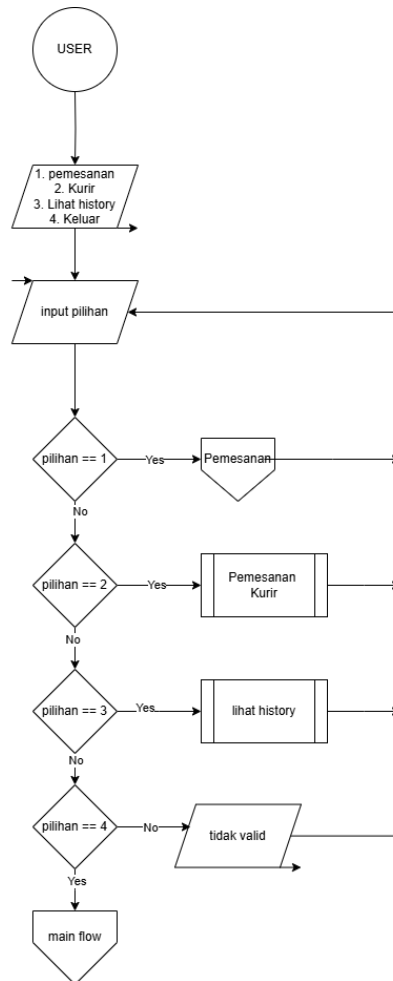
*Gambar 2.6 Flowchart Manage Layanan*

Gambar 2.6 adalah flowchart yang menjelaskan alur dari manage layanan. Ketika admin masuk ke halaman tersebut maka sistem akan menampilkan pilihan seperti lihat layanan, tambah layanan, ubah layanan, dan hapus layanan. Setelah itu admin akan diminta untuk menginputkan pilihan dimana jika admin menginputkan 1 maka sistem akan menampilkan data layanan yang terdaftar, jika admin menginputkan 2 maka sistem akan menampilkan halaman untuk menambah data layanan, jika admin menginputkan 3 maka sistem akan menampilkan halaman untuk mengubah data layanan, dan jika admin menginputkan 4 maka sistem akan menampilkan halaman yang digunakan untuk menghapus data layanan yang tersimpan. Setiap kali admin selesai melakukan proses inputan tersebut maka sistem akan kembali ke halaman admin, tetapi jika admin menginputkan angka selain angka-angka diatas maka sistem akan menganggap angka tersebut tidak valid dan meminta admin untuk kembali menginputkan angka yang valid.



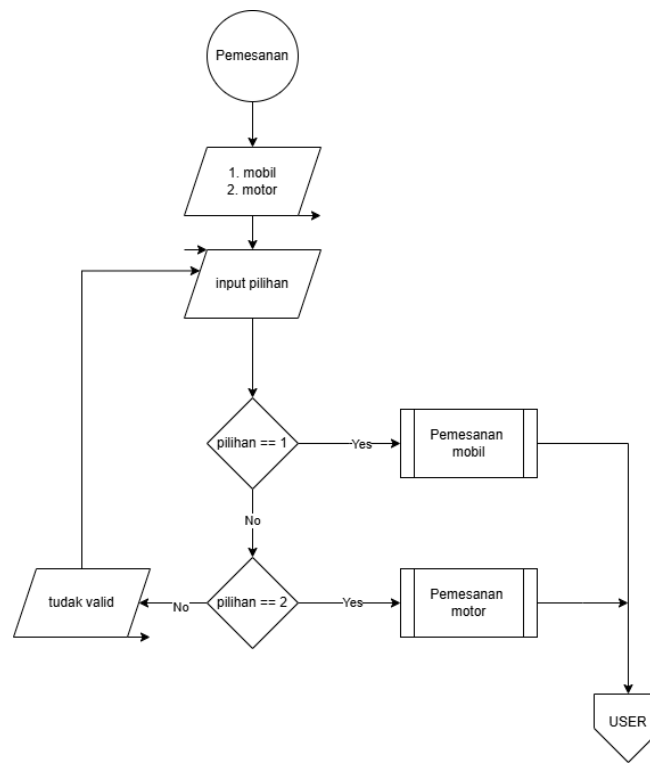
*Gambar 2.7 Manage Pesanan*

Gambar 2.7 adalah flowchart yang menjelaskan alur dari manage pesanan yang dikelola admin. Ketika admin masuk ke halaman tersebut maka sistem akan menampilkan pilihan seperti lihat pesanan, tambah pesanan, ubah pesanan, dan hapus pesanan. Setelah itu admin akan diminta untuk menginputkan pilihan dimana jika admin menginputkan 1 maka sistem akan menampilkan data pesanan yang terdaftar, jika admin menginputkan 2 maka sistem akan menampilkan halaman untuk menambah data pesanan, jika admin menginputkan 3 maka sistem akan menampilkan halaman untuk mengubah data pesanan, dan jika admin menginputkan 4 maka sistem akan menampilkan halaman yang digunakan untuk menghapus data pesanan yang tersimpan pada database. Setiap kali admin selesai melakukan proses inputan tersebut maka sistem akan kembali ke halaman admin, tetapi jika admin menginputkan angka selain angka-angka diatas maka sistem akan menganggap angka tersebut tidak valid dan meminta admin untuk kembali menginputkan angka yang valid.



*Gambar 2.8 Flowchart Menu User*

Gambar 2.8 Merupakan flowchart yang terjadi ketika user berhasil login dan masuk ke halaman user. Pada saat user masuk maka sistem akan menampilkan beberapa pilihan seperti pemesanan, kurir, lihat history, dan keluar. Setelah itu sistem akan meminta user untuk memasukkan pilihan angka dimana jika user menginputkan 1 maka sistem akan menampilkan halaman pemesanan, jika user menginputkan 2 maka sistem akan menampilkan halaman pemesanan kurir, jika user menginputkan 3 maka sistem akan menampilkan halaman history pemesanan, dan jika user menginputkan 4 maka sistem akan berakhir dan akan kembali ke halaman main flow. Tetapi jika user menginputkan angka selain angka-angka diatas maka sistem akan menampilkan tidak valid dan kembali meminta user untuk menginputkan angka yang sesuai.



*Gambar 2.9 Flowchart Pemesanan*

Gambar 2.9 Merupakan flowchart yang terjadi ketika user memilih menu pemesanan. Pada saat user masuk maka sistem akan menampilkan beberapa pilihan seperti mobil dan motor. Setelah itu sistem akan meminta user untuk memasukkan pilihan angka dimana jika user menginputkan 1 maka sistem akan menampilkan halaman pemesanan mobil dan jika user menginputkan 2 maka sistem akan menampilkan halaman pemesanan motor. Ketika user selesai melakukan proses pada masing-masing halaman tersebut maka sistem akan kembali ke halaman user. Tetapi jika user menginputkan angka selain angka-angka diatas maka sistem akan menampilkan tidak valid dan kembali meminta user untuk menginputkan angka yang sesuai.

## 2.3 Konsep/Materi Praktikum yang dipakai

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

### 1) Fungsi dasar:

1. Fungsi print digunakan secara luas untuk menampilkan keluaran di layar terminal/console. Fungsi ini berguna untuk menampilkan informasi kepada pengguna, seperti hasil input, status pesanan, kesalahan input, dan lain-lain. Misalnya, pada bagian `print(GREEN + BOLD + "Menu Admin" + RESET)` digunakan untuk menampilkan teks "Menu Admin" dengan warna tertentu untuk membedakan tampilan.
2. Fungsi input: Fungsi ini digunakan untuk menerima input dari pengguna. Misalnya, pada bagian `pilihan = input(YELLOW + "Masukkan nomor pilihan: "+RESET)`, fungsi input menerima pilihan layanan yang ingin dipesan oleh pengguna.

### 2) Struktur kontrol:

1. Percabangan (if-else): Struktur kontrol ini digunakan untuk memeriksa kondisi yang berlaku, seperti mengecek apakah input dari pengguna valid atau memeriksa peran pengguna (admin atau user) untuk menentukan akses yang sesuai. Misalnya, dalam fungsi login, program mengecek apakah username dan password yang dimasukkan cocok dengan data yang ada pada database CSV.
2. Perulangan (while-for): Struktur perulangan digunakan untuk melakukan pengecekan atau meminta input secara berulang hingga kondisi yang diinginkan tercapai. Misalnya, dalam fungsi pesan, terdapat perulangan untuk meminta input lokasi yang valid, atau dalam `menu_manage_user` untuk menampilkan pilihan menu secara terus-menerus.

### 3) Manipulasi data

Program ini menggunakan Pandas untuk membaca dan memanipulasi data yang disimpan dalam file CSV. Misalnya, untuk login dan register, program membaca file `table_user.csv` untuk memeriksa apakah username sudah terdaftar, dan untuk menyimpan data pengguna baru pada saat registrasi.

### 4) Pemrograman asinkronus

Program ini menggunakan `aiohttp` untuk mengakses API secara asynchronous, dan `asyncio` untuk menangani proses asynchronous lainnya. Ini memungkinkan program untuk melakukan beberapa operasi secara bersamaan tanpa menunggu proses sebelumnya selesai. Contohnya, dalam fungsi pesan, program memanggil API untuk mendapatkan koordinat dan menghitung jarak secara asynchronous.

### 5) Pemrograman berbasis objek dan modular

Program ini juga menggunakan pendekatan pemrograman modular, dengan membagi fungsi-fungsi besar menjadi beberapa modul yang lebih kecil dan terpisah. Contohnya, ada modul `crud.user`, `crud.pesanan`, dan `crud.layanan` yang masing-masing menangani fungsi-fungsi terkait pengelolaan data pengguna, pesanan, dan layanan.

### 6) Penanganan kesalahan

Program menggunakan pengecekan dan penanganan kesalahan untuk memastikan input pengguna valid. Misalnya, untuk memastikan bahwa input berupa angka atau memilih pilihan yang sesuai. Program juga menangani kesalahan dalam akses API dan menampilkan pesan kesalahan jika terjadi kegagalan.

### 7) Penggunaan modul dan library eksternal

#### 1. Pandas

Digunakan untuk membaca dan memanipulasi file CSV, serta mengelola data dalam bentuk tabel.

## 2. Aiohttp dan Asyncio

Digunakan untuk melakukan permintaan HTTP secara asynchronous, serta untuk mengelola proses asynchronous dalam program.

## 3. Datetime

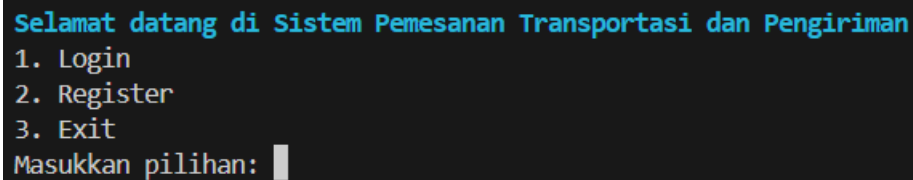
Digunakan untuk menangani waktu, seperti mencatat waktu pemesanan.

## BAB III

### HASIL DAN PEMBAHASAN

#### 3.1 Tampilan Program

##### 1. Tampilan Menu

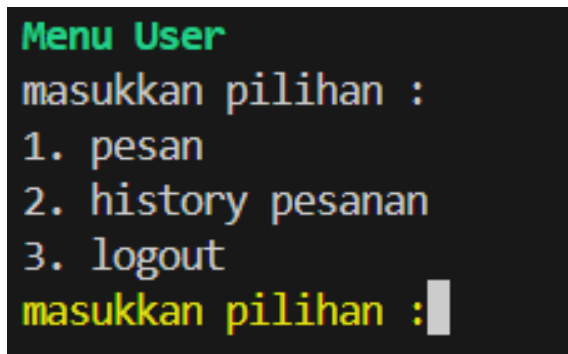


```
Selamat datang di Sistem Pemesanan Transportasi dan Pengiriman
1. Login
2. Register
3. Exit
Masukkan pilihan: █
```

*Gambar 3.1 Tampilan Menu*

Ketika pengguna memilih login maka program akan meminta user untuk menginputkan username dan password yang sudah terdaftar pada sistem. Sedangkan pada menu register, pengguna akan diminta untuk menginputkan username dan password yang akan didaftarkan kedalam sistem. Menu exit digunakan untuk keluar dari program tersebut.

##### 2. Tampilan menu user



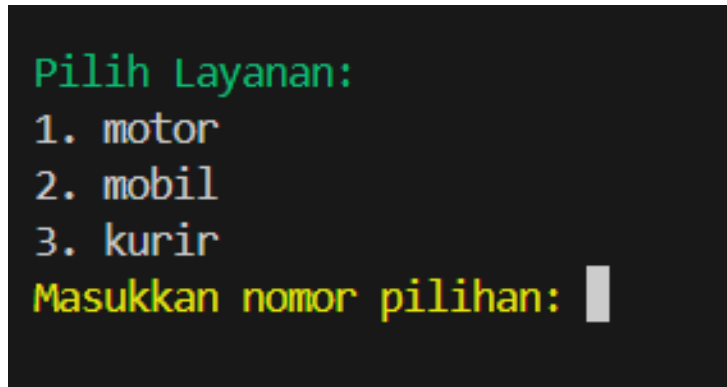
```
Menu User
masukkan pilihan :
1. pesan
2. history pesanan
3. logout
masukkan pilihan : █
```

*Gambar 3.2 Menu User*

Pada menu user terdapat 3 menu utama yaitu pesan yang digunakan pengguna untuk memilih layanan yang ingin digunakan, history pesanan yang digunakan user untuk melihat riwayat pesanan yang dilakukan dan digunakan untuk melihat status pesanan tersebut, dan menu logout digunakan pengguna untuk keluar dari menu user.

##### 3. Tampilan menu layanan





*Gambar 3.3 Menu Layanan*

Tampilan menu tersebut adalah menu yang tampil ketika pengguna memilih menu pesan pada halaman sebelumnya. Pada tampilan ini pengguna diminta untuk memilih layanan apa yang ingin digunakan, yaitu motor, mobil, atau kurir.

#### 4. Tampilan layanan motor

```
Pilih Layanan:
1. motor
2. mobil
3. kurir
Masukkan nomor pilihan: 1
Layanan yang dipilih: motor

Masukkan nama lokasi penjemputan: M yamin samarinda
M yamin samarinda ditemukan.
Praktek Dokter Umum Bersama M Yamin, Mohammad Yamin, Gunung Kelua, Samarinda Ulu, Samarinda, Kalimantan Timur, Kalimantan, 75123, Indonesia

Masukkan nama lokasi tujuan: Jl pramuka samarinda
Jl pramuka samarinda ditemukan.
Jl.Pramuka 2A, Sempaja Selatan, Samarinda Utara, Samarinda, Kalimantan Timur, Kalimantan, 75123, Indonesia

Jarak antara M yamin samarinda dan Jl pramuka samarinda adalah 1 km.
Total Harga: 5000 Rupiah.

Apakah data yang diinputkan sudah benar? (y/n): y
Pesanan telah disimpan.Silahkan menunggu pemrosesan pesanan.
```

*Gambar 3. 4 Layanan Motor*

Tampilan menu tersebut adalah tampilan yang akan tampil ketika pengguna memilih untuk menggunakan layanan motor, setelah memilih layanan motor pengguna akan diminta untuk menginputkan lokasi penjemputan dan lokasi tujuan.

#### 5. Tampilan layanan mobil

```
Pilih Layanan:
1. motor
2. mobil
3. kurir
Masukkan nomor pilihan: 2
Layanan yang dipilih: mobil

Masukkan nama lokasi penjemputan: M yamin samarinda
M yamin samarinda ditemukan.
Praktek Dokter Umum Bersama M Yamin, Mohammad Yamin, Gunung Kelua, Samarinda Ulu, Samarinda, Kalimantan Timur, Kalimantan, 75123, Indonesia

Masukkan nama lokasi tujuan: Jl pramuka samarinda
Jl pramuka samarinda ditemukan.
Jl.Pramuka 2A, Sempaja Selatan, Samarinda Utara, Samarinda, Kalimantan Timur, Kalimantan, 75123, Indonesia

Jarak antara M yamin samarinda dan Jl pramuka samarinda adalah 1 km.
Total Harga: 20000 Rupiah.

Apakah data yang diinputkan sudah benar? (y/n): y
Pesanan telah disimpan.Silahkan menunggu pemrosesan pesanan.
```

*Gambar 3. 5 Layanan Mobil*

Layanan mobil adalah tampilan yang akan tampil ketika pengguna memilih untuk menggunakan layanan mobil, sama seperti sebelumnya pada tampilan ini pengguna akan diminta untuk menginputkan lokasi penjemputan dan lokasi tujuan.

## 6. Tampilan layanan kurir

```
Pilih Layanan:
1. motor
2. mobil
3. kurir
Masukkan nomor pilihan: 3
Layanan yang dipilih: kurir

Masukkan berat barang (kg): 10
Berat barang: 10 kg

Masukkan nama lokasi penjemputan: Jl pramuka samarinda
Jl pramuka samarinda ditemukan.
Jl.Pramuka 2A, Sempaja Selatan, Samarinda Utara, Samarinda, Kalimantan Timur, Kalimantan, 75123, Indonesia

Masukkan nama lokasi tujuan: Jl perjuangan samarinda
Jl perjuangan samarinda ditemukan.
Jalan Perjuangan, Sempaja Selatan, Samarinda Utara, Samarinda, Kalimantan Timur, Kalimantan, 75119, Indonesia

Jarak antara Jl pramuka samarinda dan Jl perjuangan samarinda adalah 3 km.
Berat Barang: 10 kg.
Total Harga: 90000 Rupiah.

Apakah data yang diinputkan sudah benar? (y/n): y
Pesanan telah disimpan.Silahkan menunggu pemrosesan pesanan.
```

Gambar 3.6 Layanan Kurir

Layanan kurir adalah tampilan yang akan tampil ketika pengguna memilih untuk menggunakan layanan kurir, berbeda seperti sebelumnya pada tampilan ini pengguna akan diminta untuk menginputkan berat barang yang akan diantar, lokasi penjemputan dan lokasi tujuan.

## 7. History pesanan

```
Menu User
masukkan pilihan :
1. pesan
2. history pesanan
3. logout
masukkan pilihan :2
```

lokasi_jemput	lokasi_tujuan	jarak	layanan	beratBarang	total_harga	status	tanggal_pesanan
Jl.Pramuka	m yamin	2879	kurir	20.0	172740000	dikonfirmasi	2024-11-06 12:51:41
Jl Pramuka	M Yamin	2879	motor	-	14395000	diproses	2024-11-06 14:41:41
Jl Pramuka	M Yamin	2879	mobil	-	57580000	diproses	2024-11-06 14:52:46
Jl Pramuka	M Yamin	2879	kurir	20.0	172740000	dikonfirmasi	2024-11-06 14:58:31
Jl Pramuka samarinda	M Yamin Samarinda	2	motor	-	10000	diproses	2024-11-21 19:44:42
M yamin samarinda	Jl pramuka samarinda	1	motor	-	5000	diproses	2024-11-21 19:56:58
M yamin samarinda	Jl pramuka samarinda	1	mobil	-	20000	diproses	2024-11-21 19:57:43
Jl pramuka samarinda	Jl perjuangan samarinda	3	kurir	10.0	90000	diproses	2024-11-21 19:59:41

Tekan Enter untuk melanjutkan...

Gambar 3. 7 History Pesanan

History pesanan adalah tampilan yang akan tampil ketika pengguna memilih history pesanan, pada bagian ini pengguna dapat melihat riwayat pesanan yang telah

dilakukan. User juga dapat melihat status pesanan yang sudah dipesan pada tampilan ini.

## 8. Menu admin

```
Menu Admin
masukkan pilihan :
1. konfirmasi pesanan
2. manage user
3. manage layanan
4. manage pesanan
5. logout
masukkan pilihan : |
```

Gambar 3.8 Menu Admin

Tampilan menu admin adalah tampilan menu yang akan tampil ketika admin berhasil login. Pada tampilan ini admin dapat mengelola berbagai data seperti konfirmasi pesanan, manage user, manage layanan, manage pesanan, dan logout yang digunakan untuk keluar dari halaman tersebut.

## 9. Konfirmasi pesanan

```
Menu Admin
masukkan pilihan :
1. konfirmasi pesanan
2. manage user
3. manage layanan
4. manage pesanan
5. logout
masukkan pilihan :1
Daftar Pesanan Yang Butuh Konfirmasi:
```

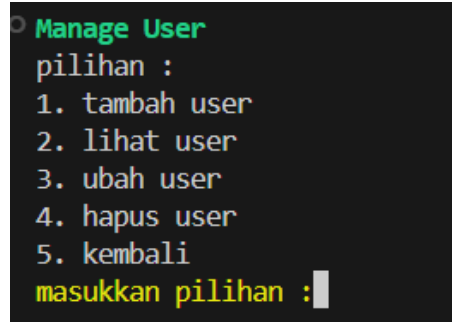
id	username	layanan	lokasi_jemput	lokasi_tujuan	jarak	beratBarang	total_harga	status
1	admin	kurir	kantor gubernur samarinda	big mall samarinda	5	5.0	75000	diproses
4	user	kurir	universitas mulawarman	umkt	4	2.0	24000	diproses
6	ghesya	motor	Jl Pramuka	M Yamin	2879	-	14395000	diproses
7	ghesya	mobil	Jl Pramuka	M Yamin	2879	-	57580000	diproses
9	ghesya	motor	Jl Pramuka samarinda	M Yamin Samarinda	2	-	10000	diproses
10	ghesya	motor	M yamin samarinda	Jl pramuka samarinda	1	-	5000	diproses
11	ghesya	mobil	M yamin samarinda	Jl pramuka samarinda	1	-	20000	diproses
12	ghesya	kurir	Jl pramuka samarinda	Jl perjuangan samarinda	3	10.0	90000	diproses

```
Pilih ID pesanan yang ingin dikonfirmasi: 11
Konfirmasi atau Tolak?
1. Konfirmasi
2. Tolak
Masukkan Pilihan: 1
Status pesanan berhasil diperbarui.
```

Gambar 3.9 Konfirmasi Pesanan

Halaman konfirmasi pesanan adalah halaman yang tampil ketika admin memilih menu konfirmasi pesanan, pada halaman ini admin akan diminta untuk menginputkan id yang ingin dikonfirmasi pesannya. Kemudian admin diminta untuk memilih menu tolak atau terima pesanan tersebut.

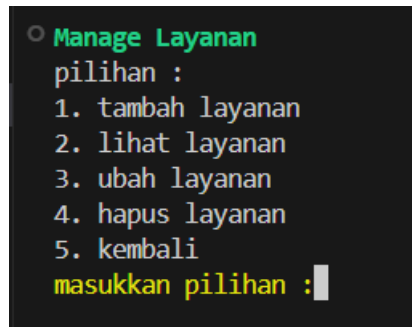
## 10. Manage user



*Gambar 3.10 Manage User*

Halaman manage user adalah menu yang akan tampil ketika user memilih manage user, halaman ini digunakan admin untuk mengelola data user seperti menambahkan, melihat, mengubah dan menghapus user. Menu kembali adalah menu yang digunakan oleh admin untuk kembali ke halaman sebelumnya.

## 11. Manage Layanan



*Gambar 3.11 Manage Layanan*

Tampilan manage layanan adalah menu yang akan tampil ketika admin memilih menu manage layanan, tampilan ini digunakan admin untuk mengelola data layanan seperti tambah layanan, lihat layanan, ubah layanan, dan hapus layanan. Sama seperti sebelumnya menu kembali adalah menu yang digunakan untuk keluar dari tampilan tersebut.

## 12. Manage pesanan

```
○ Manage pesanan
pilihan :
1. tambah pesanan
2. lihat pesanan
3. ubah pesanan
4. hapus pesanan
5. kembali
masukkan pilihan :
```

*Gambar 3.12 Manage Pesanan*

Tampilan menu manage pesanan adalah menu yang tampil ketika admin memilih menu manage pesanan pada halaman menu admin. Pada halaman ini admin dapat mengelola pesanan user seperti menambah pesanan, melihat pesanan, mengubah pesanan, dan menghapus pesanan. Sama seperti pada dua halaman menu sebelumnya, menu kembali digunakan oleh admin untuk keluar dari halaman tersebut.

### 3.2 Source Code

```
import aiohttp #untuk mengakses API (pip install aiohttp)
import asyncio #untuk menunggu proses asynchronous (pip install
asyncio)
import pandas as pd #untuk membaca file csv (pip install pandas)
import os #untuk membersihkan console (pip install os)

from datetime import datetime #untuk mendapatkan waktu

from crud.user import *
from crud.pesanan import *
from crud.layanan import *
from colors import *
from geolocation import get_jarak, get_koordinat
from invalid_pilihan import *

# fungsi login
def login(username, password):
    df = pd.read_csv('data/table_user.csv', sep=';')
    for index, row in df.iterrows():
```

```

        if row['username'] == username and row['password'] ==
password:
            data = {'id': row['id'], 'role': row['role']}
            return data
        return None

# fungsi register
def register(username, password):
    df = pd.read_csv('data/table_user.csv', sep=';')
    if username in df['username'].values:
        data = {'message' : 'Username sudah terdaftar'}
        return data

    if not df['id'].empty:
        user_id = df['id'].max() + 1
    else:

        user_id = 1 # Jika tabel kosong, mulai dengan ID 1
        role = "user"

    new_data = pd.DataFrame({
        'id': [user_id],
        'username': [username],
        'password': [password],
        'role': [role]
    })

    with open('data/table_user.csv', mode='a', newline='',
encoding='utf-8') as f:
        new_data.to_csv(f, header=False, index=False, sep=';')

    data = {'id': user_id, 'role': role, 'message': 'Pendaftaran
berhasil'}
    return data

# Fungsi Untuk Membuat Pesanan
async def pesan(user_id):
    while True:
        try:
            os.system('cls')
            print(GREEN+"Pilih Layanan:"+RESET)

```

```

df = pd.read_csv('data/table_layanan.csv', sep=';')

for idx, layanan in enumerate(df['layanan'], start=1):
    print(f"{idx}. {layanan}")

# memilih layanan
while True:
    try:
        pilihan = int(input(YELLOW+"Masukkan nomor
pilihan: "+RESET))

        # Memastikan nomor yang dipilih dalam rentang
yang valid

        if 1 <= pilihan <= len(df):
            layanan_terpilih = df.iloc[pilihan - 1]
            print(f"Layanan yang dipilih:
{layanan_terpilih['layanan']}\n")
            break
        else:
            handle_invalid_pilihan()

    except ValueError:
        handle_invalid_pilihan()

async with aiohttp.ClientSession() as session:
    berat = None

    # input berat jika jenis layanan pengiriman
    if layanan_terpilih['jenis'] == 'pengiriman':
        while True:
            try:
                berat = int(input(YELLOW + "Masukkan
berat barang (kg): " + RESET))

                if berat > 0:
                    print(f"{MAGENTA}Berat barang:
{berat} kg{RESET}\n")

                    break
                else:
                    print(RED + BOLD + "Berat bnarang
tidak boleh negatif" + RESET)

```

```

        except ValueError:
            print(RED + BOLD + "Berat barang
harus berupa angka." + RESET)

        # menentukan lokasi penjemputan
        while True:
            lokasi_jemput = input(YELLOW + "Masukkan nama
lokasi penjemputan: " + RESET)
            if not lokasi_jemput:
                print(RED + BOLD + "Lokasi tidak valid."
+ RESET)

                continue

            koordinat_jemput = await
get_koordinat(session, lokasi_jemput)
            if koordinat_jemput:
                break
            else:
                print(RED + BOLD + "Lokasi penjemputan
tidak ditemukan. Silakan coba lagi." + RESET)

        # menentukan lokasi tujuan
        while True:
            lokasi_tujuan = input(YELLOW + "Masukkan nama
lokasi tujuan: " + RESET)
            if not lokasi_tujuan:
                print(RED + BOLD + "Lokasi tidak valid."
+ RESET)

                continue

            koordinat_tujuan = await
get_koordinat(session, lokasi_tujuan)
            if koordinat_tujuan:
                break
            else:
                print(RED + BOLD + "Lokasi tujuan tidak
ditemukan. Silakan coba lagi." + RESET)

        # menentukan jarak dan menghitung total harga
jika koordinat ditemukan
        if koordinat_jemput and koordinat_tujuan:

```



```

        jarak = await get_jarak(session,
koordinat_jemput, koordinat_tujuan)

        # perhitungan total harga jika berat barang
ada
        if jarak:
            jarak = round(jarak)
            print(f"{MAGENTA}Jarak antara
{lokasi_jemput} dan {lokasi_tujuan} adalah {jarak} km.{RESET}")

            # input jarak manual jika jarak tidak
ditemukan oleh sistem
        else:
            print(RED + BOLD + "Gagal menghitung
jarak.\n" + RESET)

            pilih = input(YELLOW + "Apakah ingin
memasukkan jarak manual ? (y/n): " + RESET)
            if pilih.lower() == 'y':
                while True:
                    try:
                        jarak = float(input(YELLOW +
"Masukkan jarak (dalam satuan KM): " + RESET))
                        jarak = round(jarak)
                        break
                    except ValueError:
                        print(RED + BOLD + "Jarak
harus berupa angka." + RESET)

                elif pilih.lower() == 'n':
                    continue
                else:
                    handle_invalid_pilihan()

            # menghitung total harga
            if berat:
                total_harga = layanan_terpilih['harga'] *
jarak * berat
            else:
                total_harga = layanan_terpilih['harga'] *
jarak

```

```

        if berat:
            print(f"{MAGENTA}Berat Barang: {berat}
kg.{RESET}")

            print(f"{MAGENTA}Total Harga: {total_harga}
Rupiah.{RESET}\n")

        else:
            print("Lokasi tidak ditemukan.")

        # membuat pesanan jika jarak dan total_harga
diketahui
        if jarak and total_harga:
            while True:
                confirm = input(YELLOW + "Apakah data
yang diinputkan sudah benar? (y/n): " + RESET)
                if confirm != 'y' and confirm != 'n':
                    handle_invalid_pilihan()
                    continue
                else:
                    break

            if confirm.lower() == 'y':
                df_pesanan =
pd.read_csv('data/table_pesanan.csv', sep=';')
                if not df_pesanan['id'].empty:
                    id = df_pesanan['id'].max() + 1
                else:
                    id = 1
                # Membuat DataFrame untuk pesanan baru
                pesanan_baru = pd.DataFrame([{'id': id,
'user_id': user_id,
'lokasi_jemput': lokasi_jemput,
'lokasi_tujuan': lokasi_tujuan,
'jarak': jarak,
'layanan':
layanan_terpilih['layanan'],
'beratBarang': berat,
'total_harga': total_harga,
'status': 'diproses',

```

```

        'tanggal_pesanan':
datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    })

    with open('data/table_pesanan.csv',
mode='a', newline='', encoding='utf-8') as f:
        pesanan_baru.to_csv(f, header=False,
index=False, sep=';')

    print(GREEN+BOLD+"Pesanan telah
disimpan.Silahkan menunggu pemrosesan pesanan."+RESET)
    input("Tekan Enter untuk melanjutkan...")

    elif confirm.lower() == 'n':
        continue
    break

except KeyboardInterrupt:
    pass

# menampilkan menu admin
def manu_admin(user_id):
    while True:
        os.system('cls')
        print(GREEN + BOLD + "Menu Admin" + RESET)
        print("masukkan pilihan :")
        print("1. konfirmasi pesanan")
        print("2. manage user")
        print("3. manage layanan")
        print("4. manage pesanan")
        print("5. logout")
        pilih = input(YELLOW + "masukkan pilihan :" + RESET)

        if pilih == "1":
            konfirmasi_pesanan()
        elif pilih == "2":
            menu_manage_user(user_id)
        elif pilih == "3":
            menu_manage_layanan()
        elif pilih == "4":
            menu_manage_pesanan(user_id)

```

```

        elif pilih == "5":
            break
        else:
            handle_invalid_pilihan()

# menampilkan menu user
def menu_user(user_id):
    while True:
        try:
            os.system('cls')
            print(GREEN + BOLD + "Menu User" + RESET)
            print("masukkan pilihan :")
            print("1. pesan")
            print("2. history pesanan")
            print("3. logout")
            pilih = input(YELLOW + "masukkan pilihan :"+ RESET)

            if pilih == "1":
                asyncio.run(pesan(user_id))
            elif pilih == "2":
                history(user_id)
                input("Tekan Enter untuk melanjutkan...")
            elif pilih == "3":
                break
            else:
                handle_invalid_pilihan()
        except KeyboardInterrupt:
            pass

# menampilkan menu manage user
def menu_manage_user(user_id):
    while True:
        os.system('cls')
        print(GREEN + BOLD + "Manage User" + RESET)
        print("pilihan :")
        print("1. tambah user")
        print("2. lihat user")
        print("3. ubah user")
        print("4. hapus user")
        print("5. kembali")

```

```

pilih = input(YELLOW+ "masukkan pilihan :" + RESET)

if pilih == "1":
    while True:
        username = input("Masukkan username: ").strip()
        if username:
            break
        print(RED + "Username tidak boleh kosong" + RESET)
    while True:
        password = input("Masukkan password: ").strip()
        if password:
            break
        print(RED + "Password tidak boleh kosong" + RESET)
    while True:
        role = input("Masukkan role (admin/user):
").strip().lower()
        if role in ["admin", "user"]:
            break
        print(RED + "Role tidak valid. Masukkan 'admin' atau
'user'." + RESET)

    user = create_user(username, password, role)
    if user['status'] == "success":
        print(GREEN + user['message'] + RESET)
        input("Tekan Enter untuk melanjutkan...")
    else:
        print(RED + user['message'] + RESET)
        input("Tekan Enter untuk melanjutkan...")

elif pilih == "2":
    read_user()
    input("Tekan Enter untuk melanjutkan...")

elif pilih == "3":
    read_user()

    # Meminta ID user yang ingin diubah
    while True:
        try:
            id = int(input("Masukkan ID user: ").strip())
            if id == user_id:

```

```

        print(RED + "ID user tidak boleh sama dengan
ID yang login saat ini" + RESET)
        continue
    break
except ValueError:
    print(RED + "ID user harus berupa angka" + RESET)

    username = input("Masukkan username (kosongkan untuk
mempertahankan nilai lama): ").strip()
    password = input("Masukkan password (kosongkan untuk
mempertahankan nilai lama): ").strip()

    while True:
        role = input("Masukkan role (admin/user) atau
kosongkan untuk mempertahankan nilai lama: ").strip().lower()
        if role in ["admin", "user", ""]:
            break
        print(RED + "Role tidak valid. Masukkan 'admin' atau
'user', atau kosongkan untuk mempertahankan nilai lama." + RESET)

    user = update_user(id, username, password, role)

    if user['status'] == "success":
        print(GREEN + user['message'] + RESET)
        input("Tekan Enter untuk melanjutkan...")
    else:
        print(RED + user['message'] + RESET)
        input("Tekan Enter untuk melanjutkan...")

elif pilih == "4":
    read_user()

    # Meminta ID user yang ingin dihapus
    while True:
        try:
            id = int(input("Masukkan ID user: ").strip())
            if id == user_id:
                print(RED + "ID user tidak boleh sama dengan
ID yang login saat ini" + RESET)
                continue

```

```

        break
    except ValueError:
        print(RED + "ID user harus berupa angka" + RESET)

    user = delete_user(id)

    if user['status'] == "success":
        print(GREEN + user['message'] + RESET)
        input("Tekan Enter untuk melanjutkan...")
    else:
        print(RED + user['message'] + RESET)
        input("Tekan Enter untuk melanjutkan...")

elif pilih == "5":
    break
else:
    handle_invalid_pilihan()

# menampilkan menu manage layanan
def menu_manage_layanan():
    while True:
        os.system('cls')
        print(GREEN + BOLD + "Manage Layanan" + RESET)
        print("pilihan :")
        print("1. tambah layanan")
        print("2. lihat layanan")
        print("3. ubah layanan")
        print("4. hapus layanan")
        print("5. kembali")
        pilih = input(YELLOW+ "masukkan pilihan :" + RESET)

        if pilih == "1":
            while True:
                layanan = input("Masukkan layanan: ").strip()
                if layanan:
                    break
                print(RED + "layanan tidak boleh kosong" + RESET)
            while True:
                try:
                    harga = int(input("Masukkan harga: "))
                    if harga:

```

```

        break
        print(RED + "harga tidak boleh kosong" + RESET)
    except ValueError:
        print(RED + "harga harus berupa angka" + RESET)

    while True:
        jenis = input("Masukkan jenis
(pengiriman/transportasi): ").strip().lower()
        if jenis in ["pengiriman", "transportasi"]:
            break
        print(RED + "jenis tidak valid. Masukkan 'pengiriman'
atau 'transportasi'." + RESET)

    layanan_baru = create_layanan(layanan, harga, jenis)
    if layanan_baru['status'] == "success":
        print(GREEN + layanan_baru['message'] + RESET)
        input("Tekan Enter untuk melanjutkan...")
    else:
        print(RED + layanan_baru['message'] + RESET)
        input("Tekan Enter untuk melanjutkan...")

    elif pilih == "2":
        read_layanan()
        input("Tekan Enter untuk melanjutkan...")

    elif pilih == "3":
        read_layanan()

    # Meminta ID layanan yang ingin diubah
    while True:
        try:
            id = int(input("Masukkan ID layanan: ").strip())
            break
        except ValueError:
            print(RED + "ID layanan harus berupa angka" +
RESET)

    while True:
        layanan = input("Masukkan layanan (kosongkan untuk
mempertahankan nilai lama): ").strip()
        if layanan.isnumeric():

```



```

        print(RED + "layanan tidak boleh berupa angka" +
RESET)

        continue
    else:
        break

    while True:
        try:
            harga = int(input("Masukkan harga (masukkan 0
untuk menggunakan harga lama): ").strip())
            break
        except ValueError:
            print(RED + "harga harus berupa angka" + RESET)

    while True:
        jenis = input("Masukkan jenis
(pengiriman/transportasi) (kosongkan untuk mempertahankan nilai
lama): ").strip().lower()
        if jenis in ["pengiriman", "transportasi", ""]:
            break
        print(RED + "jenis tidak valid. Masukkan 'pengiriman'
atau 'transportasi', atau kosongkan untuk mempertahankan nilai lama."
+ RESET)

    layanan = update_layanan(id, layanan, harga, jenis)

    if layanan['status'] == "success":
        print(GREEN + layanan['message'] + RESET)
        input("Tekan Enter untuk melanjutkan...")
    else:
        print(RED + layanan['message'] + RESET)
        input("Tekan Enter untuk melanjutkan...")

elif pilih == "4":
    read_layanan()

    while True:
        try:
            id = int(input("Masukkan ID layanan: ").strip())
            break
        except ValueError:

```

```

        print(RED + "ID layanan harus berupa angka" +
RESET)

        layanan = delete_layanan(id)

        if layanan['status'] == "success":
            print(GREEN + layanan['message'] + RESET)
            input("Tekan Enter untuk melanjutkan...")
        else:
            print(RED + layanan['message'] + RESET)
            input("Tekan Enter untuk melanjutkan...")

    elif pilih == "5":
        break

# menampilkan menu Manage Pesanan
def menu_manage_pesanan(user_id):
    while True:
        os.system('cls')
        print(GREEN + BOLD + "Manage pesanan" +RESET)
        print("pilihan :")
        print("1. tambah pesanan")
        print("2. lihat pesanan")
        print("3. ubah pesanan")
        print("4. hapus pesanan")
        print("5. kembali")
        pilih = input(YELLOW+ "masukkan pilihan :" + RESET)

        if pilih == "1":
            asyncio.run(pesan(user_id))

        elif pilih == "2":
            read_pesanan()
            input("Tekan Enter untuk melanjutkan...")

        elif pilih == "3":
            read_pesanan()

        while True:
            try:
                id = int(input("Masukkan ID pesanan: ").strip())

```

```

        break
    except ValueError:
        print(RED + "ID pesanan harus berupa angka" +
RESET)

    pesanan = asyncio.run(update_pesanan(id))

    if pesanan['status'] == "success":
        print(GREEN + pesanan['message'] + RESET)
        input("Tekan Enter untuk melanjutkan...")
    else:
        print(RED + pesanan['message'] + RESET)
        input("Tekan Enter untuk melanjutkan...")

elif pilih == "4":
    read_pesanan()

    while True:
        try:
            id = int(input("Masukkan ID pesanan: ").strip())
            break
        except ValueError:
            print(RED + "ID pesanan harus berupa angka" +
RESET)

    pesanan = delete_pesanan(id)

    if pesanan['status'] == "success":
        print(GREEN + pesanan['message'] + RESET)
        input("Tekan Enter untuk melanjutkan...")
    else:
        print(RED + pesanan['message'] + RESET)
        input("Tekan Enter untuk melanjutkan...")

elif pilih == "5":
    break

else:
    handle_invalid_pilihan()

# Fungsi utama

```

```

def main():
    while True:
        # Try Except Untuk Mencegah Keyboard Interrupt
        try:
            # membersihkan console
            os.system('cls')

            print(CYAN + BOLD + "Selamat datang di Sistem Pemesanan
Transportasi dan Pengiriman" + RESET)
            print("1. Login")
            print("2. Register")
            print("3. Exit")
            pilih = input("Masukkan pilihan: ")

            if pilih == "1":

                # Input username dan password untuk login
                username = input(YELLOW + "Masukkan username:
"+RESET)

                password = input(YELLOW + "Masukkan password:
"+RESET)

                # Memeriksa apakah username dan password valid
                user = login(username, password)
                if user:
                    if user['role'] == "admin":
                        manu_admin(user['id'])
                        os.system('cls')
                    elif user['role'] == "user":
                        menu_user(user['id'])
                        os.system('cls')
                else:
                    print(RED+BOLD+"Username atau password
salah."+RESET)

                    input("Tekan Enter untuk melanjutkan...")

            elif pilih == "2":
                # Input username dan password untuk login
                username = input(YELLOW + "Masukkan username:
"+RESET)

```

```

        password = input(YELLOW + "Masukkan password:
"+RESET)

        user = register(username, password)

        if user and 'role' in user:
            print(GREEN+BOLD+user['message']+RESET)
            input("Tekan Enter untuk melanjutkan...")
            menu_user(user['id'])

        else:
            print(RED+BOLD+user['message']+RESET)
            input("Tekan Enter untuk melanjutkan...")

    elif pilih == "3":
        break

    else:
        handle_invalid_pilihan()

except KeyboardInterrupt:
    pass

if __name__ == "__main__":
    main()

```

## **BAB IV**

### **PENUTUP**

#### **4.1 Kesimpulan**

Dari laporan yang telah kami buat, kami dapat menarik kesimpulan bahwa sistem manajemen layanan pemesanan yang telah dikembangkan mampu mengelola data layanan, pengguna, dan pesanan dengan cepat dan akurat. Sistem ini dirancang untuk memberikan kemudahan bagi pengguna dalam melakukan pemesanan serta memungkinkan admin untuk mengelola data dengan lebih mudah. Tidak hanya itu, sistem juga mempunyai fitur pencatatan riwayat dan status transaksi yang memudahkan pengguna dalam melihat pesanan yang telah mereka buat.


#### **4.2 Saran**

Saran untuk pengembangan selanjutnya adalah menambahkan fitur notifikasi yang bisa lebih interaktif untuk pengguna agar pengguna dapat menerima informasi terkini mengenai status pesanan secara real-time. Tidak hanya itu, pengembang selanjutnya diharapkan melakukan pengujian lebih lanjut untuk melihat apakah sistem tetap bisa berjalan dengan baik jika digunakan oleh banyak pengguna secara sekaligus.

## DAFTAR PUSTAKA



- Charis Noijs, H., Wemaf, P. A., Nurdianty, O. A., Sohilait, W., Haumahu, S., Yusuf, H., Tomagola, K., Rindi, W., Ana, L., Salaiswa, T. S., Akuntansi, J., Ekonomi, F., & Bisnis, D. (2023). *Perancangan Sistem Informasi Akuntansi Untuk Analisis Siklus Pendapatan Pada Orantata Celullar Menggunakan DFD Dan Flowchart* (Vol. 1, Issue 2).
- Tuasamu, Z., M.Lewaru, I. A. N., Idris, R. M., Syafaat Nazari, B. A., Faradila, F., Fadlan, M., Nadiva, P., & Efendi, R. (2023). *Analisis Sistem Informasi Akuntansi Siklus Pendapatan Menggunakan DFD Dan Flowchart Pada Bisnis Porobico*.



## LAMPIRAN

<p><b>Aturan Konsultasi :</b></p> <p>A. Kartu Konsul wajib dibawa saat dilakukan konsultasi</p> <p>B. Ketua Kelompok dan Anggota diwajibkan untuk hadir tiap konsultasi</p> <p>C. Konsul dilaksanakan minimal 2 kali dengan ketentuan sebagai berikut :</p> <ul style="list-style-type: none"><li>- Konsul 1 : Konsep Program</li><li>- Konsul 2 : Penyelesaian Program</li></ul> <p>N.B : Batas waktu konsultasi ialah H+7 untuk konsul 1 dan H+14 untuk konsul 2 ( Dimulai sejak pengumuman dan pembentuk kelompok PA )</p>	<p><b>Sistem Manajemen Layanan pemesanan transportasi dan Pengiriman</b></p> <p><b>Algoritma dan Pemrograman Dasar</b></p> <p><b>KELAS AI 24:</b></p> <p><b>KELOMPOK 1:</b></p> <p>Ayu Azzahrah Alwi/ 2409106022 (Ketua)</p> <p>Ghesya Rhegta Al Rachman/ 2409106023</p> <p>Muhammad Haykal Makhmud/ 2409106005</p>  <p><b>LABORATORIUM FAKULTAS TEKNIK</b></p> <p><b>UNIVERSITAS MULAWARMAN</b></p> <p><b>2023</b></p>
---	--

*Gambar 1 Lampiran Kartu Konsul*



<p>Tanggal Konsultasi: 31 Oktober 2024</p> <p>Uraian / Pembahasan :</p> <ul style="list-style-type: none"> <li>• Flowchart off page</li> <li>• Setelah berhasil melakukan sesuatu, flowchart di buat nge loop</li> </ul>	
<p>Asisten Lab</p>  <p>Nama: Celio Arga</p>	<p>Ketua Kelompok</p>  <p>Nama: Ayu Az Zahrah Alwi</p>

<p>Tanggal Konsultasi: 7 November 2024</p> <p>Uraian / Pembahasan :</p> <ul style="list-style-type: none"> <li>• Start dan end diganti onpage off page</li> <li>• User diajarkan mengisi lokasi secara spesifik seperti nama jalan disertai dengan kotanya</li> <li>• Laporan diperbarui</li> </ul>	
<p>Asisten Lab</p>  <p>Nama: Celio Arga</p>	<p>Ketua Kelompok</p>  <p>Nama: Ayu Az Zahrah Alwi</p>

Gambar 2 Lampiran Kartu Konsul