

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



PEMESANAN TIKET KERETA API

Oleh:

Kelompok 4 C1

Muhammad Farras Arhab Ince	2409106092
Sayid Annashir Ikhwan	2409106097
Dzaki Mubarak	2409106106

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

KATA PENGANTAR

Dengan mengucapkan puji syukur ke hadirat Allah SWT, yang telah melimpahkan rahmat, taufik, dan hidayah-Nya, sehingga kami dapat menyusun laporan ini dengan baik dan tepat waktu. Laporan ini disusun sebagai salah satu bentuk tanggung jawab dalam mendokumentasikan proses dan hasil dari proyek akhir mata kuliah Algoritma dan Pemrograman Dasar, khususnya menggunakan bahasa pemrograman Python. Semoga laporan ini bermanfaat bagi pembaca dan memberikan kontribusi positif dalam pengembangan pemahaman serta keterampilan dasar pemrograman.

Dalam laporan ini, kami berusaha menyajikan informasi secara sistematis. Penyusunan laporan ini bertujuan untuk memaparkan langkah-langkah dan hasil implementasi sistem pemesanan tiket berbasis Python, termasuk analisis, desain, hingga pengujian program. Sistem ini dilengkapi fitur seperti login, pendaftaran pengguna, pengelolaan tiket, dan transaksi pemesanan tiket, sehingga mampu memberikan pengalaman simulasi pengelolaan tiket secara digital.

Kami menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun dari pembaca untuk menyempurnakan laporan ini di masa mendatang.

Ucapan terima kasih yang sebesar-besarnya kami sampaikan Mba Yuyun Nabilawati B atas bimbingan, bantuan, dan dukungannya selama pelaksanaan praktik hingga penyusunan laporan ini.

Semoga laporan ini dapat memberikan manfaat bagi pembaca

Samarinda, 18 November 2024

Kelompok 4 C1

TAKARIR

<i>Input</i>	Memasukkan
<i>Managemen</i>	Mengatur
<i>Database</i>	Basis Data
<i>Python</i>	Bahasa Pemrograman
<i>PrettyTable</i>	Table Cantik
<i>Flowchart</i>	Representasi grafis
<i>Dictionary</i>	Penyimpanan Data
<i>If</i>	Jika
<i>Else</i>	Yang Lain
<i>List</i>	Daftar
<i>Username</i>	Identitas Pengguna
<i>Password</i>	Kata Sandi
<i>Exit</i>	Keluar
<i>Admin</i>	Pengguna
<i>Print</i>	Mencetak
<i>Output</i>	Keluaran
<i>Delate</i>	Hapus
<i>Pwinput</i>	Modul python yang menampilkan ** untuk
<i>Library</i>	Perpustakaan
<i>Try</i>	Coba
<i>Except</i>	Kecuali
<i>Import</i>	Impor
<i>Break</i>	Pemutusan

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	2
1.3 Rumusan Masalah	4
1.4 Batasan Masalah.....	5
1.5 Tujuan.....	6
BAB II PERANCANGAN	7
2.1 Analisis Program	7
2.2 Flowchart	10
2.3 Konsep/Materi Praktikum yang dipakai	13
BAB III HASIL DAN PEMBAHASAN	17
3.1 Tampilan Program.....	17
3.2 Source Code	28
BAB IV PENUTUP.....	44
4.1 Kesimpulan	44
4.2 Saran.....	Error! Bookmark not defined.
DAFTAR PUSTAKA.....	46
LAMPIRAN	47

DAFTAR GAMBAR

Gambar 2.1 <i>Flowchart</i>	Error! Bookmark not defined.
Gambar 3.1 <i>Tampilan Program</i>	Error! Bookmark not defined.
Gambar 3.2 <i>Tampilan Admin</i>	Error! Bookmark not defined.
Gambar 3.3 <i>Lihat Tiket Tersedia</i>	Error! Bookmark not defined.
Gambar 3.4 <i>Tambah Tiket</i>	Error! Bookmark not defined.
Gambar 3.5 <i>Edit Tiket</i>	Error! Bookmark not defined.
Gambar 3.6 <i>Hapus Tiket</i>	Error! Bookmark not defined.
Gambar 3.7 <i>Melihat Data Pemesanan</i>	Error! Bookmark not defined.
Gambar 3.8 <i>Melihat Pengguna</i>	Error! Bookmark not defined.
Gambar 3.9 <i>Lihat Tiket Tersedia</i>	Error! Bookmark not defined.
Gambar 3.10 <i>Booking Tiket</i>	Error! Bookmark not defined.
Gambar 3.12 <i>Tiket Yang Dipesan</i>	Error! Bookmark not defined.6
Gambar 3.12 <i>Batalan Tiket</i>	Error! Bookmark not defined.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Transportasi kereta api merupakan salah satu moda transportasi yang efisien, ekonomis, dan ramah lingkungan. Dengan meningkatnya mobilitas masyarakat dalam memenuhi kebutuhan perjalanan, baik untuk urusan kerja, pendidikan, maupun liburan, sistem reservasi tiket kereta api yang cepat, akurat, dan mudah diakses menjadi kebutuhan yang sangat penting. Namun, dalam praktiknya, masih banyak ditemukan permasalahan yang menghambat kelancaran proses pemesanan tiket.

Beberapa permasalahan yang sering terjadi meliputi:

1. Antrian Panjang di Loker

Banyak calon penumpang masih mengandalkan pembelian tiket secara langsung di loket stasiun, yang sering kali menyebabkan antrian panjang dan membutuhkan waktu yang lama.

2. Kesalahan dalam Pencatatan Data

Sistem manual atau kurang terotomasi dapat meningkatkan risiko kesalahan dalam pencatatan data penumpang dan jadwal perjalanan.

3. Keterbatasan Akses

Tidak semua calon penumpang memiliki akses mudah ke loket stasiun atau sistem reservasi berbasis web, terutama bagi mereka yang berada di daerah terpencil.

4. Kurangnya Transparansi Informasi

Informasi mengenai jadwal kereta, ketersediaan tempat duduk, dan harga tiket tidak selalu mudah diakses oleh calon penumpang.

Untuk mengatasi masalah-masalah tersebut, diperlukan sebuah program aplikasi berbasis Python yang dapat membantu proses pemesanan tiket kereta api secara lebih terstruktur, cepat, dan mudah digunakan. Python dipilih karena merupakan bahasa pemrograman yang sederhana namun kuat, dengan banyak pustaka yang mendukung pengembangan aplikasi berbasis data dan antarmuka pengguna. Program ini dirancang untuk memberikan solusi praktis dengan menyediakan fitur-fitur seperti pencarian jadwal kereta, pemesanan tiket, Dengan demikian, program ini diharapkan dapat meningkatkan efisiensi dan kenyamanan bagi para pengguna jasa kereta api.

1.2 Kebutuhan Fungsional

Kebutuhan Fungsional Program Pemesanan Tiket Kereta api

1. Pengguna

- **Registrasi Akun:** Pengguna dapat mendaftar untuk membuat akun baru dengan memasukkan username dan password.
- **Login:** Pengguna dapat masuk ke akun menggunakan username dan password yang telah terdaftar.
- **Pesan Tiket Kereta api:** Pengguna dapat melihat daftar tiket kereta yang tersedia berisi stasiun, rute, harga, dan jumlah kursi
- **Lihat Tiket yang Sudah Dipesan:** Pengguna dapat melihat tiket yang telah dipesan sebelumnya.

2. Admin

- **Menampilkan Menu Admin:** Admin dapat melihat menu khusus untuk mengelola Tiket kereta api.

- Tambah Tiket: Admin dapat menambahkan Tiket baru dengan stasiun, rute, kursi, dan harga tiket.
- Lihat tiket stasiun : Admin dapat melihat daftar semua tiket-tiket stasiun yang tersedia dalam format tabel.
- Edit Tiket: Admin dapat mengedit informasi Tiket yang sudah ada (Stasiun, rute, harga, ketersediaan kursi).
- Hapus Tiket: Admin dapat menghapus tiket dari daftar stasiun yang tersedia.
- Lihat data pemesanan : Admin dapat melihat data pemesanan tiket, yang dipesan oleh pengguna

3. Interaksi Pengguna

- Tampilan Menu Utama: Program harus menampilkan menu utama yang memungkinkan pengguna untuk memilih opsi registrasi, login, atau keluar.
- Konfirmasi Aksi: Setelah melakukan pilihan maka program akan memberitahu untuk melanjutkan ke step berikutnya
- Pesan Kesalahan: Program harus memberikan pesan kesalahan jika pengguna memasukkan username atau password yang salah saat login, atau tidak sesuai dengan ketentuan

4. Data dan Penyimpanan

- Penyimpanan Data Pengguna: Program harus menyimpan data pengguna (username, password, Tiket yang dibeli) dalam struktur data yang sesuai.

- Penyimpanan Data Tiket Kereta api : Program harus menyimpan informasi tentang tiket (stasiun, rute, jumlah kursi, harga) dalam struktur data yang sesuai.

1.3 Rumusan Masalah

1. Kesulitan dalam Akses Informasi Perjalanan Kereta Api:

Banyak pengguna mengalami kesulitan dalam menemukan informasi terkini tentang jadwal kereta api, rute perjalanan, ketersediaan tempat duduk, dan harga tiket. Hal ini dapat menyebabkan kebingungan dan kehilangan kesempatan untuk memesan tiket sesuai kebutuhan.

2. Proses Pemesanan Tiket yang Rumit:

Proses pemesanan tiket kereta api sering kali rumit dan tidak efisien, terutama jika dilakukan secara manual atau melalui sistem yang tidak user-friendly. Pengguna membutuhkan sistem yang mempermudah mereka untuk memesan tiket dengan cepat dan mudah.

3. Manajemen Tiket yang Tidak Efisien:

Pengguna yang telah memesan tiket sering kesulitan mengelola dan melacak tiket yang sudah dibeli. Program ini perlu menyediakan fitur yang memungkinkan pengguna melihat detail tiket mereka dengan jelas.

4. Keterbatasan dalam Pengelolaan Jadwal dan Tiket oleh Admin:

Admin membutuhkan alat yang efektif untuk mengelola jadwal kereta api, ketersediaan tiket, dan data pemesanan. Tanpa sistem yang memadai, proses ini menjadi lebih sulit dan memakan waktu.

5. Keterbatasan dalam Fitur Registrasi dan Login:

Proses registrasi dan login yang tidak intuitif dapat menyulitkan pengguna baru. Program harus menyediakan pengalaman pengguna yang baik agar dapat menarik lebih banyak pengguna dan menjaga loyalitas pengguna yang sudah ada.

1.4 Batasan Masalah

Agar pengembangan program tetap fokus dan efisien, batasan-batasan berikut diterapkan:

1. Lingkup Wilayah:

Program hanya mencakup pemesanan tiket kereta api untuk perjalanan domestik, tidak mencakup perjalanan internasional.

2. Data Jadwal dan Rute:

Jadwal perjalanan kereta api dan rute yang digunakan bersifat simulasi dan tidak dihubungkan secara langsung dengan data resmi dari operator kereta api.

3. Jumlah Pengguna:

Program dirancang untuk penggunaan oleh satu pengguna pada satu waktu dan tidak mendukung sistem multi-user atau akses simultan.

4. Fitur Tambahan:

Program tidak mencakup fitur tambahan seperti pengelolaan pengembalian tiket, atau layanan pelanggan.

1.5 Tujuan

1. Meningkatkan Akses Informasi Perjalanan Kereta Api:

Menyediakan informasi terkini mengenai jadwal kereta, rute, harga tiket, dan ketersediaan tempat duduk agar pengguna dapat membuat keputusan perjalanan dengan tepat.

2. Meningkatkan Pengalaman Pengguna:

Mengembangkan antarmuka berbasis teks yang intuitif dan mudah digunakan, sehingga pengguna baru maupun yang sudah berpengalaman dapat dengan mudah menavigasi aplikasi dan menggunakan semua fitur yang tersedia.

3. Menyediakan Platform Pemesanan Tiket yang Efisien:

Membangun aplikasi yang memudahkan pengguna dalam melakukan pemesanan tiket kereta api secara cepat dan efisien, sehingga mereka dapat dengan mudah menemukan dan memesan tiket sesuai kebutuhan perjalanan.

4. Menyediakan Statistik Penjualan Tiket yang Berguna:

Mengimplementasikan fitur statistik yang memberikan admin informasi tentang jumlah perjalanan, tiket terjual, dan total pendapatan, sehingga mereka dapat melakukan evaluasi dan perencanaan yang lebih baik untuk pengelolaan perjalanan kereta api.

5. Mempermudah Pengelolaan Jadwal dan Tiket oleh Admin:

Mengembangkan fitur yang memungkinkan admin untuk mengelola jadwal perjalanan kereta, ketersediaan tiket, dan data pemesanan dengan mudah, termasuk menambah, mengedit, dan menghapus data, sehingga meningkatkan efisiensi operasional.

BAB II

PERANCANGAN

2.1 Analisis Program

1. Penjelasan Program:

Program ini adalah sebuah program yang dibuat untuk pemesanan tiket kereta api dimana di program ini ada dua user yaitu pengguna dan admin. admin dapat melihat tiket yang ada, menambahkan tiket, mengubah tiket, menghapus tiket, dan melihat data pembelian tiket, sedangkan pengguna dapat melihat tiket yang tersedia, booking tiket, melihat tiket dipesan, dan membatalkan tiket dipesan.

2. Alur Program:

- Menu:

1. Login
2. Daftar
3. Exit

- Login:

Jika memiliki akun, user dapat melakukan login agar dapat masuk ke Pemerograman. Tapi karena program ini memiliki dua user yaitu pengguna dan juga admin. Cara untuk Login menggunakan admin yaitu dengan memasukkan username “admin” dan password “admin123”

- Daftar:

Jika tidak memiliki akun kita dapat melakukan daftar dengan cara memasukkan username dan password baru.

- Exit:

Jika selesai menggunakan program dapat memilih ini untuk menghentikan programnya

- Menu Admin:

- | | |
|-----------------------------|-------------------|
| 1. lihat tiket tersedia | 5. data pembelian |
| 2. tambah tiket | 6. exit |
| 3. perbarui informasi tiket | |
| 4. hapus tiket | |

- Lihat tiket tersedia:

Admin dapat melihat tiket yang masih tersedia

- Tambah tiket:

Admin dapat menambahkan tiket baru dengan cara:

Memilih stasiun:

- contoh stasiun gear

Memilih rute awal:

- contoh samarinda

Rute akhir:

- contoh balikpapan

Memilih harga:

- contoh 150.000.00

Memilih jumlah tiket:

- contoh 45

Dan jika sudah di isi semua seperti contoh diatas tiket akan otomatis

Tertambah ke bagian tiket tersedia

- Perbarui Informasi Tiket:

Admin dapat mengubah data tiket yang sudah ada atau yang baru dengan cara memilih bagian mana yang kita ubah seperti stasiunnya, rute awal/akhir, harga, dan jumlah tiket tersedia

- Hapus tiket:

Admin dapat menghapus tiker yang tersedia,penhapusan ini pun bakal menghapus

- Data Pembelian:

Tiket yang sudah dipesan pengguna

- Exit:

Jika memilih ini user bakalan kembali ke menu awal

- Menu Pengguna:

1. lihat tiket tersedia
2. booking tiket
3. tiket dipesan
4. batalkan tiket
5. exit

- Lihat tiket tersedia:

Admin dapat melihat tiket yang masih tersedia

- Booking tiket:

Pengguna dapat memesan tiket yang tersedia

- Tiket dipesan:

Pengguna dapat melihat tiket yang sudah di booking

- Batalkan tiket:

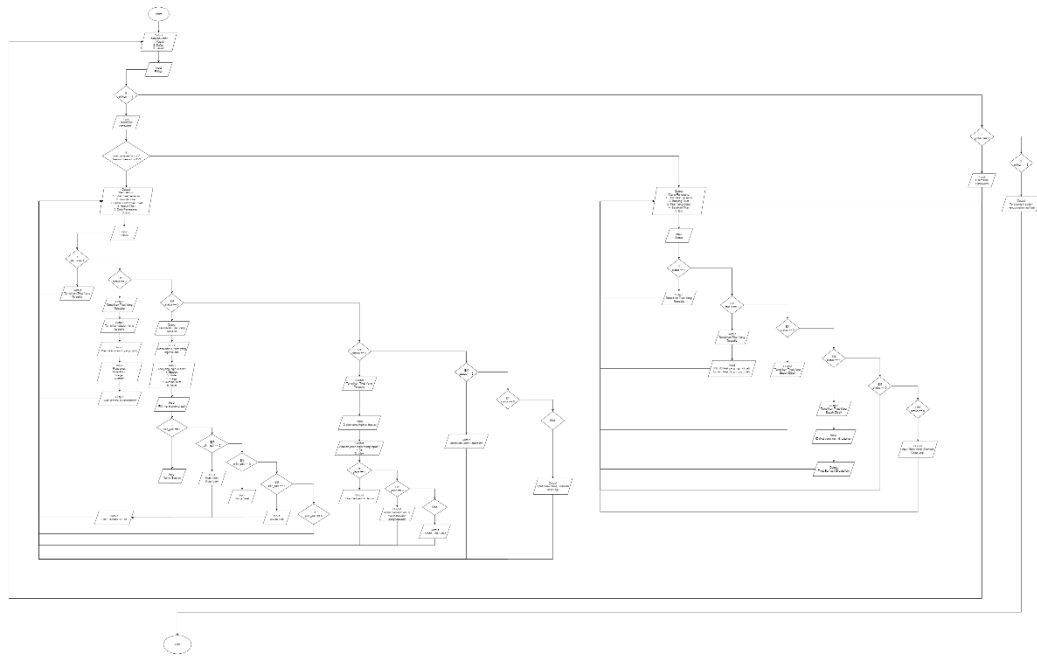
Pengguna dapat membatalkan tiket yang sudah di booking dan mengembalikan data tiket ke tiket tersedia

- Exit:

Jika memilih ini user bakalan kembali ke menu awal

2.2 Flowchart

1. Start



Gambar 2.1 Flowchart

Pertama kita dapat 3 pilihan yaitu:

1. login ,
2. daftar ,
3. Keluar,

Jika input 1, Harus memasukkan username dan password dan kita akan masuk ke programnya sebagai pengguna jika kita punya akun, jika kita memasukkan username admin dan password admin123 kita akan masuk sebagai admin

jika input 2, Harus memasukkan username dan password baru buat akun baru.

jika input 3, Bakal keluar dari program.

- Jika masuk sebagai admin kita dapat 6 pilihan yaitu

1. Lihat Tiket Tersedia,
2. Tambah Tiket,
3. Perbarui Informasi Tiket,
4. Hapus Tiket,
5. Data Pembelian,
6. Exit.

Jika input 1, Bakal melihat tiket yang tersedia.

Jika input 2, Dapat melihat tiket yang tersedia yang sudah ada dan dapat tambah tiket dengan memilih stasiun selanjut memilih rute awal dan akhir selanjutnya memilih harga dan yang terakhir memilih jumlah tiket yang tersedia.

Jika input 3, Kita dapat melihat tiket yang tersedia yang sudah ada dan yang baru ditambah, edit data tiket dengan mengubah data dari tiket yang sudah ada atau yang baru di tambah yang dipilih dan yang dapat diubah adalah stasiun, rute awal dan akhir, harga tiket, dan jumlah tiket.

Jika input 4, Dapat melihat tiket yang tersedia yang sudah ada dan yang baru ditambah, kita dapat menghapus data tiket yang sudah ada atau yang baru ditambahkan.

Jika input 5, Dapat melihat data pembelian dari pengguna.

jika input 6, Bakal kembali ke menu awal.

Jika input salah bakal muncul Input Tidak Valid, Silahkan Coba Lagi.

- Jika masuk sebagai pengguna kita dapat 5 pilihan yaitu:

1. Lihat Tiket Tersedia,
2. Beli Tiket,
3. Tiket Yang Dibeli,
4. Batalkan Tiket,
5. Exit.

Jika input 1, Kita bakal melihat tiket yang tersedia.

Jika input 2, Kita dapat melihat tiket yang tersedia yang sudah ada dan yang baru ditambah, kita dapat memilih dengan input tiket yang mau dibeli dan jumlah tiket yang mau dibeli.

Jika input 3, Kita dapat melihat data pembelian tiket yang sudah dibeli pengguna.

Jika input 4, Kita dapat melihat data pembelian tiket yang sudah dibeli pengguna, dan kita dapat menghapus data tiket yang sudah dibeli.

Jika input 5, Kita bakal kembali ke menu awal.

Jika input salah bakal muncul Input Tidak Valid, Silahkan Coba Lagi

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

Kami menggunakan fungsi dasar yaitu print yang kami gunakan untuk menampilkan Menu, Data Tiket, dan Melihatkan hasil kerja programnya kepada admin dan pengguna yang menggunakan program. Contohnya:

```
print(style.CWHITE + "\t\t Halo Selamat Datang!!")
```

2. Struktur Data

Kami menggunakan jenis data dictionary untuk data pengguna dan data tiket dengan jenis data ini kami dapat menyimpan dan mengubah seperti data tiket yang bisa dilakukan admin, dan data list untuk menyimpan username dan password akun admin. Contohnya:

```
tiket_kereta = {1: {"nama": "Stasiun Gear", "rute": "Samarinda-Balikpapan",  
"harga": 150000, "kursi": 70}}
```

```
akun_admin = ["admin", "admin123"]
```

3. Pengendalian Alur Program

Kami menggunakan **if**, **else**, dan **elif** untuk program yang kami buat karena dengan ini kami bisa membuat program bekerja seperti yang kami mau, dan dapat bekerja dari input yang dikasih user. Contoh:

```
if pilihan == 1: nama_baru = input("Masukkan nama stasiun baru: ")  
  
    tiket['nama'] = nama_baru
```

4. Fungsi

Kami membuat banyak fungsi yang terbagi bagi untuk memudahkan kami dalam membuat program, Setiap fungsinya memiliki setiap cara kerjanya seperti **def_daftar**, **def_login**, dan **def_tambah_tiket**. Contoh:

```
#Daftar akun baru  
  
def daftar():  
  
def tambah_tiket():
```

5. Penanganan Input dan Validasi

User dapat memberikan input yang dapat memperkerjakan programan supaya dapat menggunakannya, Tentu saja input harus valid dan sesuai yang ada di program supaya program dapat bekerja dengan cara melakukan validasi ke input diberikan user supaya tahu jika inputnya valid contoh jika input 1 di menu user program akan lanjut karena input nya valid karena user terdapat 5 opsi.

6. Penggunaan Library Eksternal

Kami menggunakan library eksternal Prettytable untuk menampilkan data tiket seperti id tiket, nama stasiun, rute awal/akhir, harga dan jumlah tiket tersedia dalam bentuk table hal ini membuat user dapat memudahkan pengguna untuk memesan tiket kereta api.

7. Penggunaan Variabel Global

Kami menggunakan variable global dalam program kami seperti tiket_kereta, stasiun_tersedia, dan data_pembelian. ini membuat kami dapat memanggil variable inidari dan dapat mengubah ataupun menambahkan data tiket yang tersedia di dalam program.

8. Penggunaan Class:Style

Kami menggunakan style untuk membuat programnya memiliki visual. Dimana tidak hanya teks berwarna putih tapi warna lainnya dengan menambahkan visual yang menarik supaya gak gitu aja. contoh:

```
class style():
```

```
    CRED    = '\33[31m'
```

```
    CGREEN  = '\33[32m'
```

9. Modularisasi

Program ini dirancang dengan cara yang modular, sehingga setiap bagian dari aplikasi dapat dikembangkan dan diuji secara terpisah. Ini membuat pemeliharaan dan pengembangan aplikasi lebih mudah.

10. Penggunaan Loop

Dalam program kami, kami menggunakan loop yaitu while. Dimana loop ini akan membuat program tidak berhenti sampai pengguna keluar/memberhentikan programnya dan bakal mengulang ke tempat dipasangnya loop ini jadi gak perlu ke menu awal.

BAB III HASIL DAN PEMBAHASAN

3.1 Tampilan Program

1. Tampilan Awal Program

```
=====
                Selamat datang di pemesanan tiket stasiun Mulawarman >_<
                Silahkan Pilih Menu >_<
=====
1. Masuk
2. Daftar
3. Keluar
Pilih menu anda:
```

Gambar 3.1 *Tampilan awal*

Penjelasan:

- **Masuk:**

Penjelasan:

1. Menyapa Pengguna:

- Menyapa pengguna dengan pesan sambutan dan meminta mereka untuk memasukkan username dan password.

2. Cek Kredensial:

- Fungsi memeriksa apakah username dan password yang dimasukkan cocok:
 - Jika username dan password sesuai dengan akun admin (tersimpan di variabel `akun_admin`), pengguna diberikan akses sebagai admin.

- Jika username ditemukan di dictionary pengguna dan password cocok, pengguna diberikan akses sebagai pengguna.

3. Login Berhasil:

- Jika kredensial benar:
 - Fungsi mengembalikan username, menandakan login berhasil.
 - Admin diarahkan ke menu admin.
 - Pengguna diarahkan ke menu pengguna.

4. Login Gagal:

- Jika kredensial salah:
 - Fungsi menampilkan pesan kesalahan.
 - Tidak mengembalikan nilai (None), menandakan login gagal.

- **Daftar:**

Penjelasan:

Pengguna Baru: Fungsi ini menyapa pengguna baru dan meminta mereka untuk memasukkan username untuk mendaftar.

- **Cek Username:** Fungsi memeriksa apakah username yang dimasukkan sudah ada dalam dictionary pengguna. Jika sudah ada, sistem akan memberikan pesan bahwa username tersebut sudah digunakan.
- **Masukkan Password:** Jika username belum digunakan, pengguna akan diminta untuk memasukkan password.

- **Simpan Akun:** Setelah username dan password dimasukkan, akun baru akan disimpan dalam dictionary `akun_pengguna` dengan role sebagai 'pengguna' dan tiket sebagai list kosong.
- **Konfirmasi:** Fungsi akan memberikan pesan konfirmasi bahwa akun berhasil didaftarkan.

- **Keluar:**

Penjelasan:

- **Konfirmasi Keluar:** Ketika pengguna memilih opsi "3" untuk keluar, aplikasi akan otomatis keluar
- **Pilihan Iya atau Tidak:** Pengguna diberikan dua pilihan: "1" untuk keluar dan "2" untuk tetap berada di aplikasi.
- **Keluar dari Aplikasi:** Jika pengguna memilih "1", aplikasi akan menampilkan pesan terima kasih dan keluar dari loop utama dengan perintah `break`.
- **Tetap di Aplikasi:** Jika pengguna memilih "2", aplikasi akan kembali ke menu utama.
- **Input Tidak Valid:** Jika pengguna memberikan input selain "1" atau "2", aplikasi akan memberikan pesan kesalahan dan meminta input ulang.

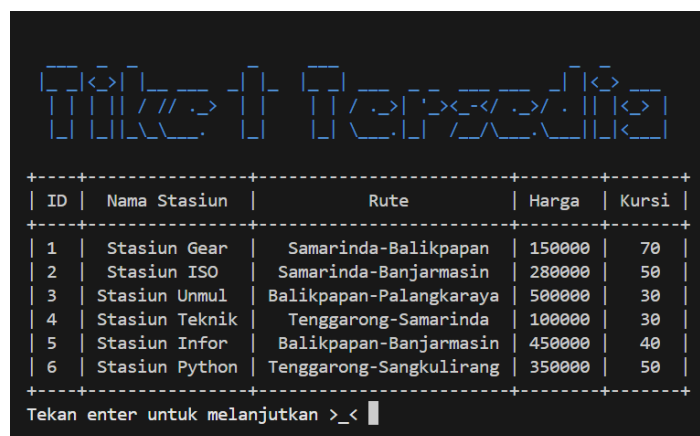
2. Tampilan Admin

```
=====
Selamat Datang Admin!
1. Lihat Tiket Tersedia
2. Tambah Tiket
3. Perbarui Informasi Tiket
4. Hapus Tiket
5. Data pemesanan
6. Exit
=====
Pilih opsi: █
```

Gambar 3.2 *Tampilan Admin*

Ketika seorang admin berhasil login, dia akan disambut dengan sebuah menu yang dirancang untuk mengelola aspek aspek pemesanan tiket kereta api. Menu ini memungkinkan admin untuk melakukan berbagai fungsi penting seperti melihat tiket tersedia, Tambah Tiket, Edit Data, Hapus Data, Data Pemesanan, dan exit.

3 . Tampilan Pilihan Menu Lihat Menu Admin



ID	Nama Stasiun	Rute	Harga	Kursi
1	Stasiun Gear	Samarinda-Balikpapan	150000	70
2	Stasiun ISO	Samarinda-Banjarmasin	280000	50
3	Stasiun Unmul	Balikpapan-Palangkaraya	500000	30
4	Stasiun Teknik	Tenggarong-Samarinda	100000	30
5	Stasiun Infor	Balikpapan-Banjarmasin	450000	40
6	Stasiun Python	Tenggarong-Sangkulirang	350000	50

Tekan enter untuk melanjutkan >_< █

Gambar 3.3 *Lihat Tiket Tersedia*

Saat memilih opsi 1 yaitu Lihat Tiket Tersedia, Kita bisa melihat ID, Nama Stadion, Rute, Harga, dan Kursi yang tersedia.

2. Tampilan Pilihan Menu Admin

```

+-----+-----+-----+-----+-----+
| ID | Nama Stasiun | Rute | Harga | Kursi |
+-----+-----+-----+-----+-----+
| 1 | Stasiun Gear | Samarinda-Balikpapan | 150000 | 70 |
| 2 | Stasiun ISO | Samarinda-Banjarmasin | 280000 | 50 |
| 3 | Stasiun Unmul | Balikpapan-Palangkaraya | 500000 | 30 |
| 4 | Stasiun Teknik | Tenggarong-Samarinda | 100000 | 30 |
| 5 | Stasiun Infor | Balikpapan-Banjarmasin | 450000 | 40 |
| 6 | Stasiun Python | Tenggarong-Sangkulirang | 350000 | 50 |
+-----+-----+-----+-----+-----+

Tekan enter untuk melanjutkan >_<

=== Tambah Tiket ===
Pilih stasiun yang tersedia:
1. Stasiun Gear
2. Stasiun ISO
3. Stasiun Unmul
4. Stasiun Teknik
5. Stasiun Infor
6. Stasiun Python
Pilih nomor stasiun: 1
Rute awal: Balikpapan
Rute akhir: Samarinda
Harga tiket (dalam IDR): 150000
Jumlah tiket tersedia: 70

Tiket berhasil ditambahkan!
Tekan enter untuk kembali ke menu admin >_<

```

Gambar 3.4 Tambah Tiket

Saat memilih opsi 2 kita bisa menambah tiket, di menu ini pertama tama kita di suruh untuk memilih stasiun yang tersedia, setelah itu kita menentukan Rute Awal, rute akhir, Harga Tiket, Serta Jumlah Tiket.

3. Tampilan Pilihan Menu Admin

```

+-----+-----+-----+-----+-----+
| ID | Nama Stasiun | Rute | Harga | Kursi |
+-----+-----+-----+-----+-----+
| 1 | Stasiun Gear | Samarinda-Balikpapan | 150000 | 70 |
| 2 | Stasiun ISO | Samarinda-Banjarmasin | 280000 | 50 |
| 3 | Stasiun Unmul | Balikpapan-Palangkaraya | 500000 | 30 |
| 4 | Stasiun Teknik | Tenggarong-Samarinda | 100000 | 30 |
| 5 | Stasiun Infor | Balikpapan-Banjarmasin | 450000 | 40 |
| 6 | Stasiun Python | Tenggarong-Sangkulirang | 350000 | 50 |
| 7 | Stasiun Gear | Balikpapan-Samarinda | 150000 | 70 |
+-----+-----+-----+-----+-----+

Tekan enter untuk melanjutkan >_<

=== Edit Data ===
Masukkan ID tiket yang ingin diedit: 7

Anda sedang mengedit tiket dengan ID 7:
Stasiun: Stasiun Gear, Rute: Balikpapan-Samarinda, Harga: 150000, Kursi Tersedia: 70

=== Apa yang anda edit? ===
1. Nama Stasiun
2. Rute
3. Harga
4. Jumlah Tiket
5. Batal
Pilih opsi: 4
Masukkan jumlah tiket baru: 80

Tiket berhasil diperbarui!
Tekan enter untuk melanjutkan >_<

```

Gambar 3.5 *Edit Tiket*

Saat memilih opsi 3 atau Edit Tiket kita diminta untuk memasukan ID berapa yang mau diedit, selanjutnya kita memilih apa yang akan kita edit, ada Nama Stasiun, Rute, Harga, Jumlah Tiket, dan Batal.

4. Tampilan Pilihan Menu Admin

```

+-----+-----+-----+-----+-----+
| ID | Nama Stasiun | Rute | Harga | Kursi |
+-----+-----+-----+-----+
| 1 | Stasiun Gear | Samarinda-Balikpapan | 150000 | 70 |
| 2 | Stasiun ISO | Samarinda-Banjarmasin | 280000 | 50 |
| 3 | Stasiun Unmul | Balikpapan-Palangkaraya | 500000 | 30 |
| 4 | Stasiun Teknik | Tenggarong-Samarinda | 100000 | 30 |
| 5 | Stasiun Infor | Balikpapan-Banjarmasin | 450000 | 40 |
| 6 | Stasiun Python | Tenggarong-Sangkulirang | 350000 | 50 |
| 7 | Stasiun Gear | Balikpapan-Samarinda | 150000 | 80 |
+-----+-----+-----+-----+
Tekan enter untuk melanjutkan >_<

=== Hapus Data ===
Masukkan ID tiket yang ingin dihapus: 7

Apakah anda yakin akan menghapus tiket?
1. Ya
2. Tidak
Hapus?: 1

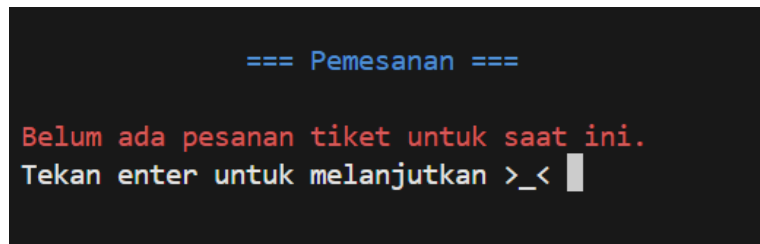
Tiket dengan ID 7 berhasil dihapus beserta data pemesanan terkait!
Tekan enter untuk melanjutkan >_<

```

Gambar 3.6 *Hapus Tiket*

Saat memilih opsi 4 kita diminta untuk memasukkan ID tiket yang ingin dihapus, setelah itu admin akan di tanyakan apakah yakin ingin menghapus tiket tersebut, pilih antara 1 atau 2, jika admin memilih 1 tiket akan terhapus, dan jika admin memilih 2 aksi tersebut tidak terjadi.

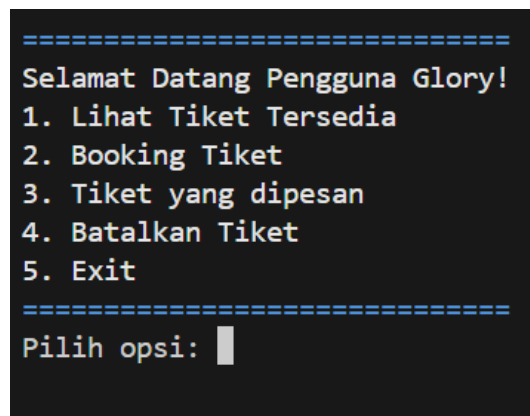
5. Tampilan Pilihan Menu Admin



Gambar 3.7 *Melihat Data Pemesanan*

Saat memilih opsi 5, admin bisa melihat data pemesanan yang sudah terjadi, admin bisa melihat Username, Nama Stasiun, Rute, dan Jumlah Tiket. Tetapi jika belum terjadi pemesanan, akan muncul pesan “Belum ada pemesanan tiket untuk saat ini.”

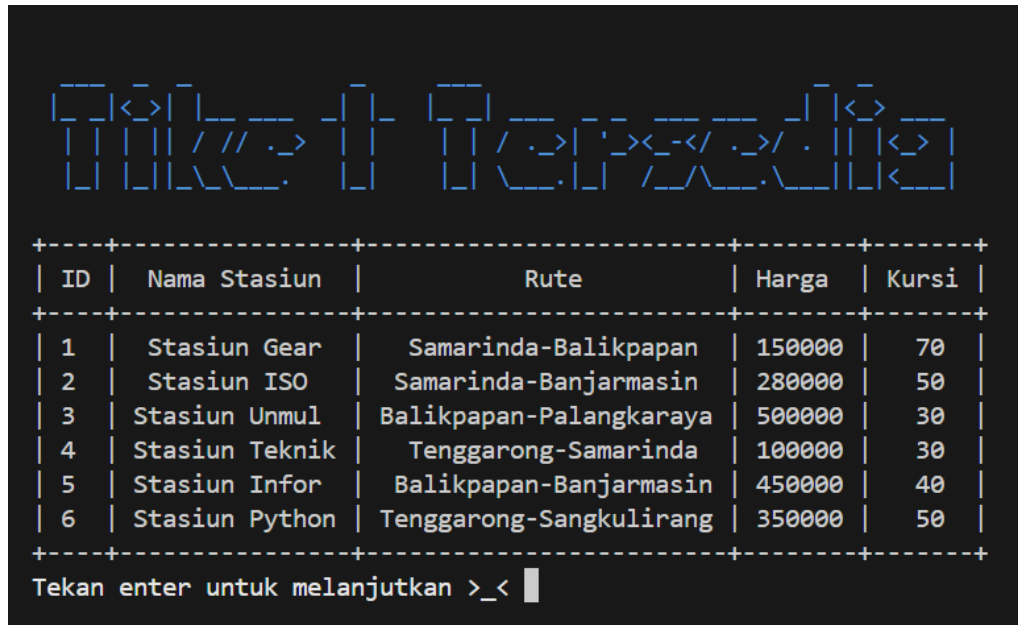
6. Tampilan Pengguna



Gambar 3.8 *Melihat Pengguna*

Ketika seorang Pengguna berhasil login, mereka akan disambut dengan sebuah menu yang dirancang untuk mengelola aspek aspek pengguna. Menu ini memungkinkan pengguna untuk Melihat Tiket Tersedia, Booking Tiket, Tiket yang di pesan, dan Batalkan Tiket.

7. Tampilan Menu Pilihan Pengguna



ID	Nama Stasiun	Rute	Harga	Kursi
1	Stasiun Gear	Samarinda-Balikpapan	150000	70
2	Stasiun ISO	Samarinda-Banjarmasin	280000	50
3	Stasiun Unmul	Balikpapan-Palangkaraya	500000	30
4	Stasiun Teknik	Tenggarong-Samarinda	100000	30
5	Stasiun Infor	Balikpapan-Banjarmasin	450000	40
6	Stasiun Python	Tenggarong-Sangkulirang	350000	50

Tekan enter untuk melanjutkan >_<

Gambar 3.9 *Lihat Tiket Tersedia*

Saat memilih opsi 1 yaitu Lihat Tiket Tersedia, Kita bisa melihat ID, Nama Stasiun, Rute, Harga, dan Kursi yang tersedia.

8. Tampilan Menu Lihat Pengguna

```

+-----+-----+-----+-----+
| ID | Nama Stasiun | Rute | Harga | Kursi |
+-----+-----+-----+-----+
| 1 | Stasiun Gear | Samarinda-Balikpapan | 150000 | 70 |
| 2 | Stasiun ISO | Samarinda-Banjarmasin | 280000 | 50 |
| 3 | Stasiun Unmul | Balikpapan-Palangkaraya | 500000 | 30 |
| 4 | Stasiun Teknik | Tenggarong-Samarinda | 100000 | 30 |
| 5 | Stasiun Infor | Balikpapan-Banjarmasin | 450000 | 40 |
| 6 | Stasiun Python | Tenggarong-Sangkulirang | 350000 | 50 |
+-----+-----+-----+-----+

Tekan enter untuk melanjutkan >_<

=== Booking Tiket ===
Masukkan ID tiket yang ingin dipesan: 1
Masukkan jumlah tiket yang ingin dipesan: 2

Tiket berhasil dipesan! Total: 2 tiket di Stasiun Gear.
Tekan enter untuk melanjutkan >_<

```

Gambar 3.10 *Booking Tiket*

Saat memilih opsi 2 kita dapat memesan tiket yang ingin dipesan, dengan cara memilih ID tiket serta memasukkan jumlah tiket yang ingin dipesan

9. Tampilan Menu Lihat Pengguna

```

=== Tiket yang Dipesan oleh Glory ===
+-----+-----+-----+-----+
| ID Tiket | Nama Stasiun | Rute | Jumlah Tiket |
+-----+-----+-----+-----+
| 1 | Stasiun Gear | Samarinda-Balikpapan | 2 |
+-----+-----+-----+-----+

Tekan enter untuk melanjutkan >_<

```

Gambar 3.11 *Tiket Yang Dipesan*

Saat memilih opsi 3, pengguna dapat melihat tiket yang sudah ia pesan, dengan menampilkan ID Tiket, Nama Stasiun, Rute, dan Jumlah Tiket yang dipesan. Jika pengguna belum memesan apapun akan muncul pesan “Anda belum memesan tiket apa pun.”

10. Tampilan Menu Lihat Pengguna

```
=== Tiket yang Dipesan oleh Glory ===
+-----+-----+-----+-----+
| ID Tiket | Nama Stasiun | Rute | Jumlah Tiket |
+-----+-----+-----+-----+
| 1 | Stasiun Gear | Samarinda-Balikpapan | 2 |
+-----+-----+-----+-----+
Tekan enter untuk melanjutkan >_<

=== Batalkan Tiket ===
Masukkan ID tiket yang ingin dibatalkan: 1

Tiket berhasil dibatalkan dan jumlah kursi diperbarui!
Tekan enter untuk melanjutkan >_< █
```

Gambar 3.12 Batalkan Tiket

Saat memilih opsi 4, pengguna dapat membatalkan tiket yang sudah di pesan, dengan cara memilih ID Tiket yang ingin di batalkan.

3.2 Source Code

```
# Fitur Menghapus Terminal agar terlihat rapih
import os
# Fitur Mmembuat Table
# os.system("pip install PrettyTable")
from prettytable import PrettyTable
# Fitur Membuat ketika input password jadi bintang ****
# os.system("pip install pwinput")
import pwinput

class style():
    CRED      = '\33[31m'
    CGREEN    = '\33[32m'
    CWHITE    = '\33[37m'
    CYELLOW   = '\33[33m'
    CBLUE     = '\33[34m'
    ENDC      = '\033[0m'

# Data pengguna dan tiket kereta
pengguna = {}

# Menyimpan username dan password admin
akun_admin = ["admin", "admin123"]

# Tiket Kereta Yang Tersedia Dari Awal menggunakan nested dictionary
tiket_kereta = {
    1: {"nama": "Stasiun Gear", "rute": "Samarinda-Balikpapan",
        "harga": 150000, "kursi": 70},
    2: {"nama": "Stasiun ISO", "rute": "Samarinda-Banjarmasin",
        "harga": 280000, "kursi": 50},
    3: {"nama": "Stasiun Unmul", "rute": "Balikpapan-Palangkaraya",
        "harga": 500000, "kursi": 30},
    4: {"nama": "Stasiun Teknik", "rute": "Tenggarong-Samarinda",
        "harga": 100000, "kursi": 30},
    5: {"nama": "Stasiun Infor", "rute": "Balikpapan-Banjarmasin",
        "harga": 450000, "kursi": 40},
    6: {"nama": "Stasiun Python", "rute": "Tenggarong-Sangkulirang",
        "harga": 350000, "kursi": 50}
}

#stasiun yang tersedia
```



```

hapus()
print(style.CBLUE + "=" * 35)
print(style.CWHITE + "\tHalo pengguna baru!")
print("\t\tSilahkan Daftar")
print(style.CBLUE + "=" * 35)
username_baru = input(style.CWHITE + "Masukkan Username: ")
# Cek apakah username sudah ada atau digunakan untuk admin
if username_baru in pengguna or username_baru == akun_admin[0]:
    print(style.CRED + "\nUsername sudah digunakan!" + style.ENDC)
else:
    password_baru = pwininput.pwininput(style.CWHITE + "Masukkan
Password: ")
    if len(password_baru) < 8:
        print(style.CRED + "\nPassword tidak boleh kurang dari 8
karakter." + style.ENDC)
    else:
        pengguna[username_baru] = {'password': password_baru,
'role': 'pengguna', 'tiket': []}
        print(style.CGREEN + "\nPendaftaran berhasil! Anda
terdaftar sebagai pengguna.")

    input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)

# Menu Masuk
def login():
    hapus()
    print(style.CBLUE + "=" * 45)
    print(style.CWHITE + "\t\t\tHalo Selamat Datang!!")
    print("\t\t\tSilahkan Login")
    print(style.CBLUE + "=" * 45)
    username = input(style.CWHITE + "Username: ")
    password = pwininput.pwininput(style.CWHITE + "Kata Sandi: " +
style.ENDC)

    # Cek apakah username dan password cocok dengan akun admin
    if username == akun_admin[0] and password == akun_admin[1]:
        return akun_admin[0] # Kembalikan username admin jika cocok

    # Jika bukan admin, cek dalam data pengguna biasa

```

```

        elif username in pengguna and pengguna[username]['password'] ==
password:
            return username
        else:
            print(style.CRED + "\nUsername atau kata sandi salah!")

            input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)

# Tampilan Admin
def tampilan_admin():
    hapus()
    print(style.CBLUE + "=" * 30)
    print(style.CWHITE + f"Selamat Datang Admin!")
    print("1. Lihat Tiket Tersedia\n2. Tambah Tiket\n3. Perbarui
Informasi Tiket\n4. Hapus Tiket\n5. Data pemesanan\n6. Exit")
    print(style.CBLUE + "=" * 30 + style.ENDC)

# Tampilan Pengguna
def tampilan_pengguna(username):
    hapus()
    print(style.CBLUE + "=" * 30)
    print(style.CWHITE + f"Selamat Datang Pengguna {username}!")
    print("1. Lihat Tiket Tersedia\n2. Booking Tiket\n3. Tiket yang
dipesan\n4. Batalkan Tiket\n5. Exit")
    print(style.CBLUE + "=" * 30 + style.ENDC)

# Fungsi Melihat tiket yang tersedia (data dari dictionary
'tiket_kereta')
def lihat_tiket():
    hapus()
    # Table yang bagian atas
    table = PrettyTable(["ID", "Nama Stasiun", "Rute", "Harga",
"Kursi"])
    # Table bagian isi
    for ID_Tiket, info in tiket_kereta.items():
        table.add_row([ID_Tiket, info["nama"], info["rute"],
info["harga"], info["kursi"]])
    # Menampilkan Table
    logo_tiket_tersedia()
    print(table)

```

```

        input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)

# Fungsi dari menu admin untuk menambah tiket yang tersedia
def tambah_tiket():
    hapus()
    # Tampilkan Table
    lihat_tiket()
    print(style.CBLUE + "\n=== Tambah Tiket ===" + style.ENDC)

    # Pilih stasiun
    print(style.CYELLOW + "Pilih stasiun yang tersedia:" + style.ENDC)
    # Menampilkan stasiun yang sudah berada di dictionary
'tiket_kereta'
    for i, stasiun in enumerate(stasiun_tersedia, start=1):
        print(style.CWHITE + f"{i}. {stasiun}" + style.ENDC)
    try:
        pilihan_stasiun = int(input(style.CWHITE + "Pilih nomor
stasiun: " + style.ENDC)) - 1
        # Mengecek apakah pilihan stasiun berada di data
stasiun_tersedia
        if 0 <= pilihan_stasiun < len(stasiun_tersedia):
            # pilihan stasiun akan disimpan di variable nama_stasiun
            nama_stasiun = stasiun_tersedia[pilihan_stasiun]
        else:
            print(style.CRED + "\nPilihan tidak valid. Silakan coba
lagi." + style.ENDC)
            input(style.CWHITE + "Tekan enter untuk kembali ke menu
admin >_< " + style.ENDC)
            return
        except ValueError:
            print(style.CRED + "\nInput harus berupa angka. Silakan coba
lagi." + style.ENDC)
            input(style.CWHITE + "Tekan enter untuk kembali ke menu admin
>_< " + style.ENDC)
            return

    # Minta rute awal dan rute akhir
    rute_awal = input(style.CWHITE + "Rute awal: " + style.ENDC)
    rute_akhir = input(style.CWHITE + "Rute akhir: " + style.ENDC)
    # Menyimpan rute awal dan rute akhir di variable 'rute'

```

```

rute = f"{rute_awal}-{rute_akhir}"

# Input harga
try:
    harga = int(input(style.CWHITE + "Harga tiket (dalam IDR): " +
style.ENDC))
    if harga <= 0:
        print(style.CRED + "\nHarga harus lebih dari 0. Silakan
coba lagi." + style.ENDC)
        input(style.CWHITE + "Tekan enter untuk kembali ke menu
admin >_< " + style.ENDC)
        return
    except ValueError:
        print(style.CRED + "\nInput harus berupa angka. Silakan coba
lagi." + style.ENDC)
        input(style.CWHITE + "Tekan enter untuk kembali ke menu admin
>_< " + style.ENDC)
        return

# Input jumlah tiket
try:
    kursi = int(input(style.CWHITE + "Jumlah tiket tersedia: " +
style.ENDC))
    if kursi <= 0:
        print(style.CRED + "\nJumlah tiket tidak boleh dibawah
angka 1. Silakan coba lagi." + style.ENDC)
        input(style.CWHITE + "Tekan enter untuk kembali ke menu
admin >_< " + style.ENDC)
        return
    elif kursi > 80:
        print(style.CRED + "\nJumlah Tiket Tersedia tidak boleh
lebih dari 80." + style.ENDC)
        input(style.CWHITE + "Tekan enter untuk kembali ke menu
admin >_< " + style.ENDC)
        return
    else:
        # Menentukan ID tiket yang baru ditambahkan jika kursi yang
di input sudah benar, dengan mengecek keys di tiket_kereta maxnya
berapa
        id_tiket_baru = max(tiket_kereta.keys()) + 1 if
tiket_kereta else 1

```

```

        # Menambahkan tiket di dictionary tiket_kereta
        tiket_kereta[id_tiket_baru] = {"nama": nama_stasiun,
"route": rute, "harga": harga, "kursi": kursi}
        print(style.CGREEN + "\nTiket berhasil ditambahkan!" +
style.ENDC)
        input(style.CWHITE + "Tekan enter untuk kembali ke menu
admin >_< " + style.ENDC)
        return
    except ValueError:
        print(style.CRED + "\nInput harus berupa angka. Silakan coba
lagi." + style.ENDC)
        input(style.CWHITE + "Tekan enter untuk kembali ke menu admin
>_< " + style.ENDC)
        return

# Fungsi dari menu pengguna untuk booking
def booking_tiket(username):
    lihat_tiket()
    try:
        # memilih id tiket yang ingin dipesan
        print(style.CBLUE + "\n=== Booking Tiket ===" + style.ENDC)
        ID_Tiket = int(input("Masukkan ID tiket yang ingin dipesan: "))
        if ID_Tiket not in tiket_kereta:
            print(style.CRED + "\nID tiket tidak valid!" + style.ENDC)
            input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)
            return # kembali ke menu pengguna

        # memasukkan jumlah tiket yang ingin dipesan
        jumlah = int(input("Masukkan jumlah tiket yang ingin dipesan:
"))
        if jumlah <= 0:
            print(style.CRED + "\nJumlah tiket harus lebih dari 0!" +
style.ENDC)
            input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)
            return # kembali ke menu pengguna

        # Mengecek apakah jumlah tiket yang tersedia lebih besar atau
sama dengan jumlah tiket yang diminta.

```

```

        if tiket_kereta[ID_Tiket]['kursi'] >= jumlah:
            # Mengurangi Jumlah Tiket yang Tersedia akan mengubah data
            di dictionary tiket_kereta.
            tiket_kereta[ID_Tiket]['kursi'] -= jumlah
            # menambahkan data pemesanan
            data_pemesanan.append({
                "username": username,
                "nama_stasiun": tiket_kereta[ID_Tiket]["nama"],
                "rute": tiket_kereta[ID_Tiket]["rute"],
                "jumlah_tiket": jumlah,
                "ID_Tiket": ID_Tiket
            })
            print(style.CGREEN + f"\nTiket berhasil dipesan! Total:
{jumlah} tiket di {tiket_kereta[ID_Tiket]['nama']}." + style.ENDC)
            input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)
        else:
            print(style.CRED + f"\nJumlah tiket yang diminta melebihi
kursi yang tersedia ({tiket_kereta[ID_Tiket]['kursi']})." + style.ENDC)
            input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)
        except ValueError:
            print(style.CRED + "\nInput tidak valid! Harap masukkan angka."
+ style.ENDC)
            input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)

# Fungsi Menu admin untuk Melihat Data Pemesanan
def lihat_data_pemesanan():
    hapus()
    # Mengecek apakah sudah ada pemesanan yang terjadi
    print(style.CBLUE + "\n
                    === Pemesanan ===" +
style.ENDC)
    if not data_pemesanan:
        print(style.CRED + "\nBelum ada pesanan tiket untuk saat ini."
+ style.ENDC)
        input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)
    else:
        # Membuat table Pemesanan

```



```

        table = PrettyTable(["Username", "Nama Stasiun", "Rute",
"Jumlah Tiket"])
        for pemesanan in data_pemesanan:
            table.add_row([pemesanan["username"],
pemesanan["nama_stasiun"], pemesanan["rute"],
pemesanan["jumlah_tiket"]])
            # print tamble pemesanan
            print(table)
            input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)

#Fungsi Menu admin untuk mengedit data tiket_kereta
def edit_data():
    hapus()
    lihat_tiket()

    print(style.CBLUE + "\n=== Edit Data ===" + style.ENDC)
    try:
        # memilih ID tiket yang ingin di ediy]t
        id_tiket = int(input("Masukkan ID tiket yang ingin diedit: "))
        # Mengecek apakah ada id tiket di dictionary tiket_kereta
        if id_tiket not in tiket_kereta:
            print(style.CRED + "\nID tiket tidak valid!" + style.ENDC)
            input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)
            return

        # Menyimpan pilihan id_tiket di variable 'tiket'
        tiket = tiket_kereta[id_tiket]
        # Memberitahu admin id tiket apa yang akan dia edit
        print(style.CYELLOW + f"\nAnda sedang mengedit tiket dengan ID
{id_tiket}:" + style.ENDC)
        print(f"Stasiun: {tiket['nama']}, Rute: {tiket['rute']}, Harga:
{tiket['harga']}, Kursi Tersedia: {tiket['kursi']}")

        # Pilihan untuk mengedit
        print(style.CBLUE + "\n=== Apa yang anda edit? ===" +
style.ENDC)
        print("1. Nama Stasiun")
        print("2. Rute")
        print("3. Harga")

```

```

print("4. Jumlah Tiket")
print("5. Batal")

pilih_edit = int(input("Pilih opsi: "))

# Edit Nama Stasiun
if pilih_edit == 1:
    nama_baru = input("Masukkan nama stasiun baru: ")
    tiket['nama'] = (f"Stasiun {nama_baru}")
# Edit Rute
elif pilih_edit == 2:
    rute_awal = input("Masukkan rute awal baru: ")
    rute_akhir = input("Masukkan rute akhir baru: ")
    tiket['rute'] = f"{rute_awal}-{rute_akhir}"
# Edit Harga
elif pilih_edit == 3:
    harga_baru = int(input("Masukkan harga tiket baru: "))
    tiket['harga'] = harga_baru
# Edit Jumlah Tiket
elif pilih_edit == 4:
    kursi_baru = int(input("Masukkan jumlah tiket baru: "))
    if kursi_baru > 80:
        print(style.CRED + "\nJumlah Tiket Tersedia tidak boleh
lebih dari 80." + style.ENDC)
        input(style.CWHITE + "Tekan enter untuk kembali ke menu
admin >_< " + style.ENDC)
        return
    else:
        tiket['kursi'] = kursi_baru
#Batal Edit
elif pilih_edit == 5:
    print(style.CGREEN + "\nPerubahan dibatalkan." +
style.ENDC)
    input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)
    return
else:
    print(style.CRED + "\nPilihan tidak valid." + style.ENDC)
    input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)
    return

```

```

        print(style.CGREEN + "\nTiket berhasil diperbarui!" +
style.ENDC)

    except ValueError:
        print(style.CRED + "\nInput tidak valid." + style.ENDC)

    input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)

# Fungsi Hapus Data
def hapus_data():
    hapus()
    lihat_tiket()

    print(style.CBLUE + "\n=== Hapus Data ===" + style.ENDC)
    try:
        id_tiket = int(input("Masukkan ID tiket yang ingin dihapus: "))
        if id_tiket in tiket_kereta:
            print(style.CYELLOW + "\nApakah anda yakin akan menghapus
tiket?" + style.ENDC)
            print("1. Ya\n2. Tidak")
            yakin = int(input("Hapus?: "))

            if yakin == 1:
                # Hapus tiket dari tiket_kereta
                del tiket_kereta[id_tiket]
                # Hapus tiket dari data pemesanan
                global data_pemesanan
                # Melihat apakah ID_tiket di data_pemesanan sama dengan
id_tiket(yg ingin dihapus) jika iya, tiket tersebut akan di hapus
                data_pemesanan = [pemesanan for pemesanan in
data_pemesanan if pemesanan["ID_Tiket"] != id_tiket]
                print(style.CGREEN + f"\nTiket dengan ID {id_tiket}
berhasil dihapus beserta data pemesanan terkait!" + style.ENDC)
                input(style.CWHITE + "Tekan enter untuk melanjutkan >_<
" + style.ENDC)
                return
            elif yakin == 2:
                print(style.CGREEN + "\nAnda memilih untuk batalkan
penghapusan." + style.ENDC)

```

```

        input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " + style.ENDC)
        return
    else:
        print(style.CRED + "\nPilihan tidak valid." + style.ENDC)
        input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " + style.ENDC)
        return
    else:
        print(style.CRED + "\nID tiket tidak valid!" + style.ENDC)
        input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " + style.ENDC)

except ValueError:
    print(style.CRED + "\nInput tidak valid." + style.ENDC)
    input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " + style.ENDC)

# Fungsi Melihat Pemesanan yang sudah di pesan oleh pengguna
def tiket_dipesan(username):
    hapus()
    print(style.CBLUE + f"=== Tiket yang Dipesan oleh {username} ===" + style.ENDC)
    # Mengecek apakah sudah ada pemesanan yang terjadi
    if not data_pemesanan:
        print(style.CRED + "\nAnda belum memesan tiket apa pun." + style.ENDC)
        input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " + style.ENDC)
    else:
        # Membuat tabel untuk menampilkan tiket yang dipesan
        table = PrettyTable(["ID Tiket", "Nama Stasiun", "Rute", "Jumlah Tiket"])
        # Menambahkan semua tiket yang dipesan oleh pengguna ke tabel
        for pemesanan in data_pemesanan:
            # Mengecek apakah nilai username di dalam data_pemesanan cocok dengan username yang dimiliki oleh pengguna yang sedang login.
            if pemesanan["username"] == username:
                # Tambahkan setiap pemesanan yang sesuai dengan username ke dalam tabel

```

```

        table.add_row([pemesanan["ID_Tiket"],
pemesanan["nama_stasiun"], pemesanan["rute"],
pemesanan["jumlah_tiket"]])

    print(table)
    input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)

# Fungsi Menu Pengguna Untuk membatalkan tiket yang sudah di pesan
def batalkan_tiket(username):
    hapus()
    # Tampilkan tiket yang dipesan oleh pengguna
    tiket_dipesan(username)

    try:
        # Meminta ID tiket yang ingin dibatalkan
        print(style.CBLUE + "\n=== Batalkan Tiket ===" + style.ENDC)
        id_tiket = int(input(style.CWHITE + "Masukkan ID tiket yang
ingin dibatalkan: " + style.ENDC))

        # Memeriksa apakah ID tiket ada di dalam data pemesanan
pengguna
        tiket_dipesan_oleh_pengguna = [pemesanan for pemesanan in
data_pemesanan if pemesanan["username"] == username and
pemesanan["ID_Tiket"] == id_tiket]

        if not tiket_dipesan_oleh_pengguna:
            print(style.CRED + "\nTiket yang Anda pilih tidak ada dalam
pemesanan Anda." + style.ENDC)
            input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)
            return

        # Jika tiket ditemukan, batalkan pemesanan
        for pemesanan in tiket_dipesan_oleh_pengguna:
            # Menambahkan kembali jumlah kursi yang dibatalkan ke tiket
yang tersedia
            tiket_kereta[pemesanan["ID_Tiket"]]["kursi"] +=
pemesanan["jumlah_tiket"]
            # Menghapus pemesanan dari data_pemesanan

```

```

        data_pemesanan.remove(pemesanan)

        print(style.CGREEN + "\nTiket berhasil dibatalkan dan jumlah
kursi diperbarui!" + style.ENDC)
        input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)

    except ValueError:
        print(style.CRED + "\nInput tidak valid! Harap masukkan angka."
+ style.ENDC)
        input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)

while True:
    tampilan_awal()

    try:
        pilihan = int(input("Pilih menu anda: "))
    except ValueError:
        print(style.CRED + "\nInput tidak valid. Tolong Masukkan Angka
1/2/3." + style.ENDC)
        input(style.CWHITE + "Tekan enter untuk melanjutkan >_< " +
style.ENDC)
        continue

    if pilihan == 1:
        username = login()

        if username:
            while True:
                if username == akun_admin[0]:
                    tampilan_admin()
                    try:
                        status = int(input("Pilih opsi: "))
                    except ValueError:
                        print(style.CRED + "\nInput tidak valid.
Silakan masukkan angka yang sesuai." + style.ENDC)
                        input(style.CWHITE + "Tekan enter untuk
melanjutkan >_< " + style.ENDC)
                        continue

```

```

        if status == 1:
            lihat_tiket()
        elif status == 2:
            tambah_tiket()
        elif status == 3:
            edit_data()
        elif status == 4:
            hapus_data()
        elif status == 5:
            lihat_data_pemesanan()
        elif status == 6:
            break
        else:
            print(style.CRED + "\nPilihan tidak valid, coba
lagi." + style.ENDC)
            input(style.CWHITE + "Tekan enter untuk
melanjutkan >_< ")
    else:
        tampilan_pengguna(username)
        try:
            status = int(input("Pilih opsi: "))
        except ValueError:
            print(style.CRED + "\nInput tidak valid.
Silakan masukkan angka yang sesuai.")
            input(style.CWHITE + "Tekan enter untuk
melanjutkan >_< ")
            continue

    if status == 1:
        lihat_tiket()
    elif status == 2:
        booking_tiket(username)
    elif status == 3:
        tiket_dipesan(username)
    elif status == 4:
        batalkan_tiket(username)
    elif status == 5:
        break
    else:

```

```
        print(style.CRED + "\nPilihan tidak valid, coba  
lagi." + style.ENDC)  
        input(style.CWHITE + "Tekan enter untuk  
melanjutkan >_< ")  
    elif pilihan == 2:  
        daftar()  
    elif pilihan == 3:  
        goodbye()  
        break  
    else:  
        print(style.CRED + "\nPilihan tidak valid, coba lagi." +  
style.ENDC)  
        input(style.CWHITE + "Tekan enter untuk melanjutkan >_< ")
```


BAB IV PENUTUP

4.1 Kesimpulan

Program pemesanan tiket kereta api yang dikembangkan oleh Kelompok 4 C1 berhasil dirancang dan diimplementasikan sebagai simulasi sistem yang mempermudah pengguna dan admin dalam mengelola tiket perjalanan. Program ini memberikan solusi terhadap masalah yang sering terjadi dalam sistem manual, seperti kesalahan pencatatan data, keterbatasan akses informasi, dan antrian panjang di loket. Dengan menggunakan Python sebagai bahasa pemrograman, program ini dilengkapi fitur registrasi pengguna, login, pemesanan tiket, pembatalan tiket, serta pengelolaan data tiket oleh admin. Implementasi pustaka PrettyTable memberikan keunggulan dalam penyajian data, sehingga mempermudah pengguna dalam memahami informasi tiket yang tersedia. Proses validasi input juga memastikan bahwa program dapat berjalan dengan stabil dan bebas dari kesalahan fatal.

Meskipun telah memenuhi kebutuhan dasar sistem pemesanan tiket, program ini masih memiliki beberapa keterbatasan, seperti tidak adanya fitur pengembalian tiket atau dukungan untuk multi-user. Namun, keberhasilan proyek ini memberikan kontribusi signifikan dalam pemahaman dan penguasaan dasar-dasar algoritma, struktur data, serta pengembangan aplikasi berbasis teks.

4.2 Saran

Agar program dapat melakukan transaksi pembelian tiket secara lebih lengkap, disarankan untuk menambahkan fitur integrasi dengan sistem pembayaran elektronik. Integrasi ini memungkinkan pengguna menyelesaikan pembayaran langsung melalui aplikasi dengan menggunakan metode seperti e-

wallet atau transfer bank. Selain itu, modul khusus untuk menangani transaksi dapat dirancang, mencakup fungsi perhitungan total biaya tiket, pembuatan invoice, dan konfirmasi pembayaran setelah metode pembayaran dipilih.

Penyimpanan data transaksi perlu diperhatikan dengan menyimpan informasi penting seperti data tiket, metode pembayaran, tanggal transaksi, dan status pembayaran. Data ini berguna untuk melacak riwayat pembelian pengguna dan mengelola laporan keuangan. Fitur notifikasi transaksi juga dapat ditambahkan untuk memberi tahu pengguna mengenai status transaksi mereka, misalnya, pesan konfirmasi setelah pembayaran berhasil dilakukan.

Selain itu, validasi pada pembayaran harus diterapkan guna menghindari kesalahan seperti duplikasi pembayaran atau pembayaran yang tidak sesuai dengan harga tiket. Untuk tahap awal pengembangan, dapat disertakan simulasi sistem pembayaran sederhana agar fitur ini dapat diuji tanpa memerlukan layanan eksternal. Penambahan pelaporan transaksi bagi admin juga penting untuk memberikan informasi terkait rekapitulasi penjualan tiket, membantu pengelolaan sistem menjadi lebih transparan dan efisien.

DAFTAR PUSTAKA

Zainuddin, M. (2021). *Pemrograman Python untuk Pemula*. Yogyakarta: Andi Offset.

Kusnadi, R. (2020). *Dasar-Dasar Pemrograman Python*. Jakarta: Elex Media Komputindo.

Wahyu, S. (2022). *Algoritma dan Struktur Data dengan Python*. Bandung: Informatika.

Sumber Online/Referensi Web:

Dokumentasi Resmi Python

<https://docs.python.org/3/>

Tutorial Python PrettyTable

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

Real Python - Python Tutorials

<https://realpython.com/>

GeeksforGeeks - Python Programming

<https://www.geeksforgeeks.org/python-programming-language/>

AI dan Chatbot Referensi:

ChatGPT (OpenAI)

<https://chat.openai.com/>

LAMPIRAN

Tiket Kereta Api Algoritma dan Pemrograman Dasar



KELAS Informatika C1'24:
KELOMPOK 4:

Muhammad Farras Arhab Ince / 2409106092(ketua)
Sayid Annashir Ikhwani / 2409106097
Dzaki Muhabarak / 24091060106



LABORATORIUM FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
2024

Tanggal Konsultasi : 01 November 2024	
Uraian / Pembahasan : • Output jalurnya ke End Soran • Password jadi bintang saat ditulis • Password dibuat jadi batas (karakter)	
Asisten Lab	Ketua Kelompok
 Nama: Yugen Nabila P.	 Nama: M. Farras Arhab Ince

Tanggal Konsultasi : 13 November 2024	
Uraian / Pembahasan : • Tambah Stasiun Menjadi 6 Stasiun • Kasih limit kursi yang di tambahkan • Tambahkan Warna • Tambah pesan "Belum ada pesanan" di lihat data • Tambah hapus data - pembelian • Flowchart tambahkan Elif dan Else	
Asisten Lab	Ketua Kelompok
 Nama: Yugen Nabila P.	 Nama: M. Farras Arhab Ince