**LAPORAN UJIAN AKHIR SEMESTER I**

**APLIKASI SEDERHANA SISTEM PEMESANAN TIKET KERETA**



Disusun Oleh :

Nama : Cintya Della Permatasari

NPM : 5220411199

Mata Kuliah : Algoritma Pemrograman Teori D

Dosen Pengampu :

**Moh. Ali Romli, S.Kom., M.Kom.**

**PROGRAM STUDI INFORMATIKA FAKULTAS SAINS & TEKNOLOGI UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**2022/2023**

# **KATA PENGANTAR**

Assalamu’alaikum warahmatullahi wabarakatuh

Segala puji syukur kita panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penyusun dapat menyelesaikan laporan tepat waktu. Laporan ini memberikan referensi ilmu yang baru untuk dunia akademis. Alasan inilah yang mendorong penulis memilih untuk membuat laporan ini

Saya selaku penulis sangat berterima kasih kepada dosen pengampu mata kuliah Algoritma Pemrograman Teori yaitu Bapak Moh. Ali Romli, S.Kom., M.Kom. yang telah memberikan serta menjelaskan materi selama satu semester ini sehingga saya dapat menyelesaikan laporan ini.

Tujuan penulis membuat laporan ini adalah untuk memenuhi tugas individu Ujian Akhir Semester I pada mata kuliah Algoritma Pemrograman Teori. Maka dari itu penulis membuat laporan dengan judul “Aplikasi Sederhana Sistem Pemesanan Kereta Api.”

Saya berharap laporan yang saya buat ini memberikan manfaat kepada pembaca. Saya menyadari laporan yang saya buat ini memiliki banyak kekurangan. Oleh karena itu, saya sebagai penyusun dengan rendah hati menerima apabila ada kritik maupun saran yang membangun demi kesempurnaan laporan ini.

Wassalamu’alaikum warahmatullahi wabarakatuh

Yogyakarta, 18 Januari 2023

CINTYA DELLA PERMATASARI

# **DAFTAR ISI**

[**KATA PENGANTAR** 2](#_Toc124936696)

[**DAFTAR ISI** 3](#_Toc124936697)

[**BAB I** 5](#_Toc124936698)

[**PENDAHULUAN** 5](#_Toc124936699)

[**1.** **Latar Belakang** 5](#_Toc124936700)

[**2.** **Identifikasi Masalah** 5](#_Toc124936701)

[**3.** **Batasan Masalah** 5](#_Toc124936702)

[**4.** **Tujuan** 5](#_Toc124936703)

[**BAB II** 6](#_Toc124936704)

[**LANDASAN TEORI** 6](#_Toc124936705)

[**1.** **Sistem** 6](#_Toc124936706)

[**2.** **Aplikasi** 6](#_Toc124936707)

[**3.** **Database** 6](#_Toc124936708)

[**4.** **PHP (Perl Hypertext Preprocesor)** 6](#_Toc124936709)

[**5.** **MySQL** 6](#_Toc124936710)

[**BAB III** 7](#_Toc124936711)

[**PEMBAHASAN** 7](#_Toc124936712)

[**1.** **FLOWCHART** 7](#_Toc124936713)

[**2.** **ANALISIS SISTEM BERJALAN** 8](#_Toc124936714)

[**3.** **IMPLEMENTASI DAN PENGUJIAN** 8](#_Toc124936715)

[**1.** **Membuat Database** 8](#_Toc124936716)

[**2.** **Membuat Tabel** 9](#_Toc124936717)

[**3.** **Koneksi Database** 9](#_Toc124936718)

[**4.** **Tampilan Menu Utama** 9](#_Toc124936719)

[**5.** **Tampilan Login sebagai Admin** 10](#_Toc124936720)

[**6.** **Daftar Pemesan** 11](#_Toc124936721)

[**7.** **Cari Data Pemesan** 11](#_Toc124936722)

[**8.** **Grafik Penjualan** 12](#_Toc124936723)

[**9.** **Login sebagai Pemesan** 12](#_Toc124936724)

[**10.** **Menu Pemesanan Kereta Api** 13](#_Toc124936725)

[**11.** **Menu Lihat Data Pemesan** 15](#_Toc124936726)

[**12. Menu Pembayaran** 16](#_Toc124936727)

[**13. Menu Ubah Data Pemesan** 18](#_Toc124936728)

[**14. Menu Pembatalan** 21](#_Toc124936729)

[15. Menu Cetak Ulang Bukti Pembayaran 22](#_Toc124936730)

[**BAB IV** 22](#_Toc124936731)

[**PENUTUP** 22](#_Toc124936732)

[**KESIMPULAN** 22](#_Toc124936733)

[**SARAN** 22](#_Toc124936734)

[**DAFTAR PUSTAKA** 23](#_Toc124936735)

# **BAB I**

# **PENDAHULUAN**

## **Latar Belakang**

Saat ini penerapan teknologi semakin berkembang, semua aktivitas yang dilakukan oleh sebuah bidang usaha tidak terlepas dari pengaruh teknologi. Dalam sebuah bisnis, customer merupakan faktor yang sangat penting. Untuk itu, diperlukan membina suatu hubungan yang baik antara perusahaan dengan customer. Dengan mempelajari perilaku customer dan keinginan customer, diharapkan perusahaan dapat mengambil langkah-langkah yang tepat untuk meningkatkan pelayanan kepada customer, meningkatkan loyalitas customer, serta meningkatkan pendapatan perusahaan. Meningkatan pelayanan terhadap customer dapat dilakukan dengan beberapa cara, antara lain dengan memberikan fasilitas-fasilitas tertentu kepada customer atau dengan memberikan informasi yang cepat, akurat dan memiliki jangkauan yang luas.

Maka dari itu perlu dibuat suatu aplikasi pemesanan tiket kereta api yang dimana dalam aplikasi tersebut diharapkan mampu memberikan informasi kepada customer mengenai nama kereta api, jenis kelas, kota asal, kota tujuan, jadwal keberangkatan dan tiba serta harga. Berdasarkan permasalahan yang ada maka penulis tertarik untuk mengambil judul “Aplikasi Sederhana Sistem Pemesanan Kereta Api” dengan menggunakan bahasa pemrograman Python dan database MySQL.

## **Identifikasi Masalah**

Berdasarkan latar belakang di atas, adapun identifikasi masalah adalah sebagai berikut :

1. Bagaimana membangun aplikasi yang mampu memberikan informasi dan juga melakukan pemesanan ke server.
2. Bagaimana menghubungkan bahasa pemrograman Python dengan database MySQL.
3. Bagaimana merancang dan membuat sebuah aplikasi sistem pemesanan kereta api yang dapat menampilkan informasi tentang kereta api, mulai dari nama kereta api, jenis kelas, kota asal, kota tujuan, jadwal keberangkatan dan tiba serta harga.
4. Bagaimana merancang suatu aplikasi sistem sederhana yang mudah digunakan.

## **Batasan Masalah**

Dalam suatu penelitian diperlukan adanya batasan masalah agar dalam penelitian tersebut dapat terarah. Beberapa batasan masalah yang digunakan diantaranya sebagai berikut :

1. Analisis, perancangan, serta implementasi yang dibahas dalam sistem mencangkup informasi kereta, kota keberangkatan dan tujuan , serta jadwal keberangkatan dan tiba.
2. Aplikasi memberikan informasi perjalanan kereta kelas ekonomi, eksekutif dan bisnis.

## **Tujuan**

Adapun tujuan dari pembuatan aplikasi ini adalah sebagai berikut :

1. Membuat aplikasi sistem pemesanan tiket kereta api dengan menggunakan bahasa pemrograman python dan database MySQL.
2. Merancang suatu sistem aplikasi sederhana yang mudah digunakan.
3. Laporan pemesanan dapat disajikan lebih cepat, tepat, dan akurat.

# **BAB II**

# **LANDASAN TEORI**

## **Sistem**

Menurut Herlambang (2005), definisi dari sistem dapat dibagi menjadi duas pendekatan, yaitu pendekatan secara prosedur dan pendekatan secara komponen. Berdasarkan pendekatan prosedur, sistem didefinisikan sebagai sekumpulan dari beberapa prosedur yang memiliki suatu tujuan tertentu. Sedangkan berdasarkan pendekatan komponen, sistem merupakan kumpulan dari komponen yang saling berkaitan untuk mencapai suatu tujuan tertentu.

## **Aplikasi**

Aplikasi merupakan sebuah software (perangkat lunak) yang bertugas sebagai front end pada sebuah sistem yang dipakai untuk mengelola berbagai macam data sehingga dapat menjadi sebuah informasi yang bermanfaat untuk pengguna dan sistem.

Aplikasi yang akan dibuat adalah suatu program yang menyediakan fungsi untuk aktivasi admin yaitu melihat dan mencari data pemesan sedangkan pemesan melakukan pemesanan, pembayaran, pengubahan data dan pembatalan pemesanan tiket kereta api.

Aplikasi yang berupa software adalah suatu program yang dibuat khusus untuk memudahkan pengguna dalam melakukan suatu kegiatan sesuai dengan kebutuhan.

## **Database**

Menurut Marlinda (2004) database adalah suatu susunan atau komponen data operasional lengkap dari suatu organisasi atau peusahaan yang dikelola dan disimpan secara terintegrasi denggan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan oleh pengguna.

## **PHP (Perl Hypertext Preprocesor)**

PHP adalah singkatan dari (Perl Hypertext Preprocesor) yaitu bahasa pemograman web server-side yang bersifat open source. PHP merupakan script yang terintegrasi dengan HTML dan berada pada server (server side HTML embedded scripting). PHP adalah script yang digunakan untuk membuat halaman website yang dinamis.

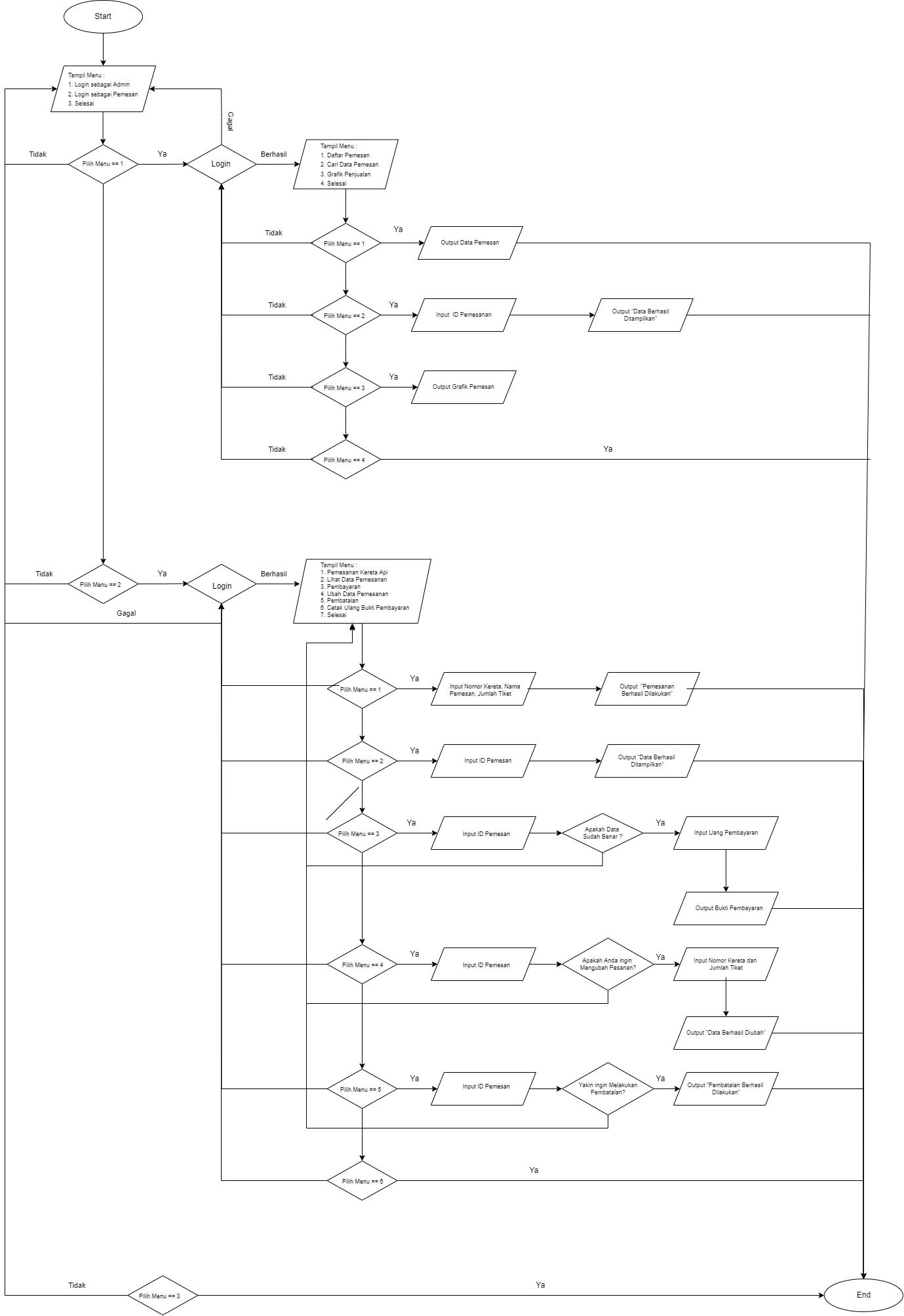
## **MySQL**

Menurut Budi Raharjo (2015) MySQL adalah suatu RDBMS ( server database ) yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak pengguna. MySQL sebagai salah satu software untuk database server yang banyak digunakan, MySQL bersifat Open Source dan biasa dijalankan diberbagai platform misalnya windows Linux, dan lain sebagainya.

# **BAB III**

# **PEMBAHASAN**

## **FLOWCHART**



## **ANALISIS SISTEM BERJALAN**

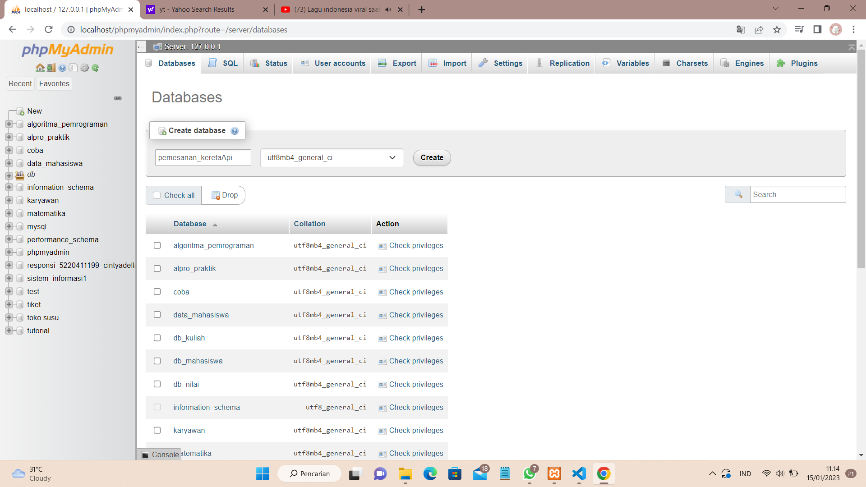
Aplikasi yang saya buat merupakan Aplikasi Sederhana Sistem Pemesanan Kereta Api yang dibuat menggunakan bahasa pemrograman python dan database MySQL. Terdapat 5 library yang digunakan untuk merancang aplikasi ini diantaranya : pretty table, pandas, matplotlib, datetime dan os. Aplikasi Sederhana Sistem Pemesanan Kereta Api dapat digunakan untuk melakukan pemesanan tiket kereta api.

Cara kerja aplikasi ini adalah yang pertama user perlu melakukan login terlebih dahulu, ada 2 pilihan login yaitu login sebagai admin dan login sebagai pemesan. Login sebagai admin digunakan untuk pegawai yang dimana dalam login sebagai admin sudah ditentukan untuk username dan passwordnya, jadi apabila salah menginputkan maka tidak bisa lanjut ke menu selanjutnya. Namun, apabila login berhasil maka akan ditujukan ke menu selanjutnya yaitu menu untuk melihat daftar pemesan, mencari data pemesan dan melihat grafik penjualan.

Sedangkan jika memilih login sebagai pemesan, user tidak perlu menginputkan ussername dan password karena pada menu login ini dapat diakses oleh semua orang. Setelah login, maka user akan disajikan beberapa pilihan menu diantaranya : menu untuk melakukan pemesanan, lihat data, pembayaran, ubah data, pembatalan dan menu untuk mencetak ulang bukti pembayaran.

### **IMPLEMENTASI DAN PENGUJIAN**

### **Membuat Database**



Langkah pertama untuk membuat Aplikasi Sederhana Sistem Pemesanan Tiket Kereta Api adalah membuat database terlebih dahulu di phpMyAdmin.

### **Membuat Tabel**

Sebuah gambar berisi teks, cuplikan layar, putih

Deskripsi dibuat secara otomatis

Setelah database berhasil dibuat kemudian membuat tabel, isi tabel diisi sesuai dengan kebutuhan. Pada kolom Ide saya menggunakan auto increment dengan tujuan agar kode bisa didapatkan secara otomatis dan nantinya bisa digunakan untuk primary key.

### **Koneksi Database**

|  |  |
| --- | --- |
| Sebuah gambar berisi teks, monitor, cuplikan layar, layar  Deskripsi dibuat secara otomatis  Source code untuk mengkoneksikan python ke database, pada source code ini menggunakan import mysql.connector agar python dapat terhubung ke database. | Apabila di running dan database berhasil terkoneksi maka pada hasil running akan muncul seperti gambar diatas. |

### **Tampilan Menu Utama**

|  |  |
| --- | --- |
| Sebuah gambar berisi teks, monitor, cuplikan layar, elektronik  Deskripsi dibuat secara otomatis  Pada source code menu utama menggunakan fungsi perulangan while. Apabila sebuah kondisi terpenuhi maka program akan terus melakukan perulangan, namun apabila suatu kondisi tidak terpenuhi maka perulangan akan berhenti. | Hasil running tampilan menu utama pada Aplikasi Pemesanan Tiket Kereta Api. |

### **Tampilan Login sebagai Admin**

|  |
| --- |
| Sebuah gambar berisi teks, monitor, cuplikan layar, layar  Deskripsi dibuat secara otomatis |

Source code pilihan menu admin menggunakan fungsi perulangan while. Apabila sebuah kondisi terpenuhi maka program akan terus melakukan perulangan, namun apabila suatu kondisi tidak terpenuhi maka perulangan akan berhenti. Terdapat juga fungsi perulangan if else dimana pada fungsi tersebut apabila data yang diinputkan memenuhi fungsi if maka akan langsung di eksekusi namun apabila data yang diinput tidak sesuai akan dialihkan ke kondisi kedua dan seterusnya. Namun apabila data yang diinputkan tidak memenuhi semua kondisi maka akan dialihkan ke kondisi else dimana proses perulangan akan berhenti dan tidak ada data yang di eksekusi. Selain fungsi perulangan terdapat juga library os yang fungsinya untuk membersihkan layar pada program yang akan dijalankan sehingga pemrogram dapat menenentukan saat untuk menghapus data yang telah dijalankan tanpa harus menutup program tersebut dan membukanya kembali.

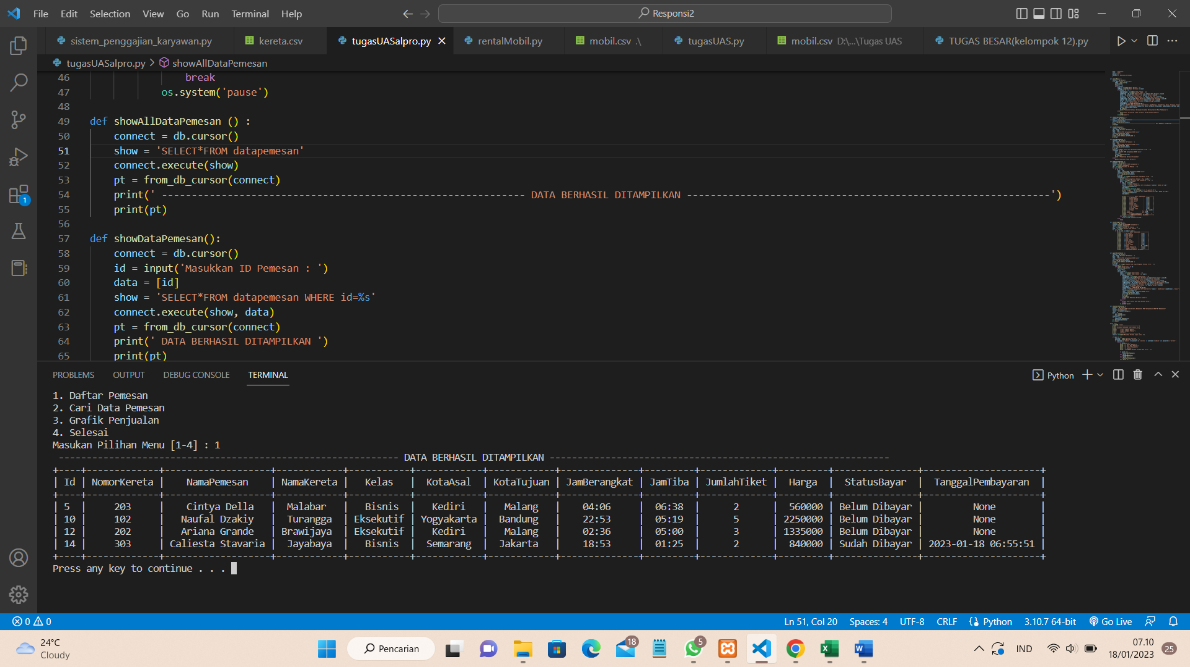
|  |  |
| --- | --- |
| Hasil running pada menu login sebagai admin. Apabila username dan password yang diinputkan salah maka akan muncul pernyataan bahwa username atau password salah dan pengguna diminta untuk menginputkan kembali username dan password yang benar. | Sebuah gambar berisi teks, monitor, cuplikan layar, layar  Deskripsi dibuat secara otomatis  Apabila username dan password yang dimasukan benar maka akan muncul menu login sebagai admin.  Sebuah gambar berisi teks, monitor, cuplikan layar, layar  Deskripsi dibuat secara otomatis  Pada menu login sebagai admin terdapat 4 pilihan menu. |

### **Daftar Pemesan**

Sebuah gambar berisi teks, monitor, cuplikan layar, layar

Deskripsi dibuat secara otomatis

Source code untuk menampilkan seluruh data pemesan pada Aplikasi Pemesanan Kereta Api. Pada source code ini terdapat library prettytable yang gunanya untuk membuat tabel hasil running yang lebih terstruktur dan rapi.



Hasil running akan menampilkan semua data pemesan yang berada di dalam database tanpa terkecuali.

### **Cari Data Pemesan**

|  |  |
| --- | --- |
|  | Source code untuk mencari data dari pemesan, opsi menu ke 2 ini juga terdapat dalam pilihan menu login sebagai pemesan. Dalam source code untuk melihat data dari salah satu pemesan terdapat fungsi library pretty table yang gunanya untuk membuat tabel hasil running yang lebih terstruktur. |

Sebuah gambar berisi teks, monitor, cuplikan layar, layar

Deskripsi dibuat secara otomatis

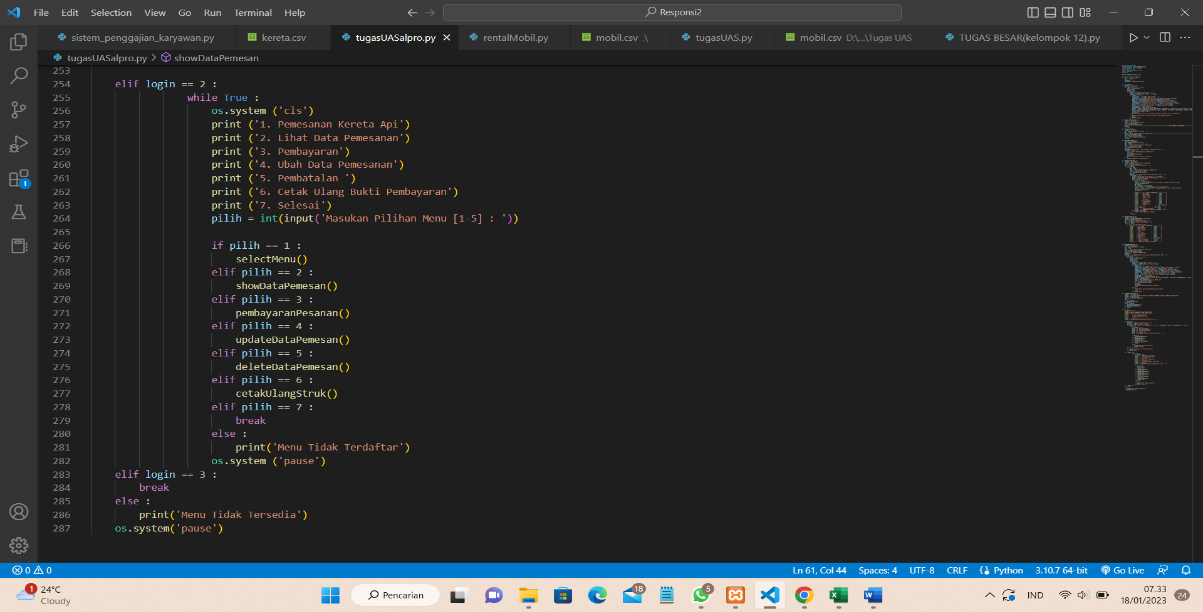
Hasil running menu ke 2 yaitu cari data pemesan, admin perlu menginputkan ID Pemesan agar dapat menampilkan data dari salah satu pemesan.

### **Grafik Penjualan**

|  |  |
| --- | --- |
|  |  |

Source code untuk menampilkan grafik penjualan. Grafik yang ditampilkan merupakan grafik jumlah penjualan tiket kereta api berdasarkan dari nama kereta api.

### **Login sebagai Pemesan**



Source kode pilihan menu admin menggunakan fungsi perulangan while. Apabila sebuah kondisi terpenuhi maka program akan terus melakukan perulangan, namun apabila suatu kondisi tidak terpenuhi maka perulangan akan berhenti. Terdapat juga fungsi perulangan if else dimana pada fungsi tersebut apabila data yang diinputkan memenuhi fungsi if maka akan langsung di eksekusi namun apabila data yang diinput tidak sesuai akan dialihkan ke kondisi kedua dan seterusnya. Namun apabila data yang diinputkan tidak memenuhi semua kondisi maka akan dialihkan ke kondisi else dimana proses perulangan akan berhenti dan tidak ada data yang di eksekusi. Selain fungsi perulangan terdapat juga library os yang fungsinya untuk membersihkan layar pada program yang akan dijalankan sehingga pemrogram dapat menenentukan saat untuk menghapus data yang telah dijalankan tanpa harus menutup program tersebut dan membukanya kembali.

|  |  |
| --- | --- |
|  | Hasil running source code login 2 akan menampilkan menu untuk pemesanan kereta api, nelihat data pemesan, pembayaran, mengubah data pemesan, pembatalan, cetak bukti pembayaran dan terakhir menu selesai yaitu menu untuk kembali ke halaman menu utama. |

### **Menu Pemesanan Kereta Api**

def selectMenu () :

  connect = db.cursor()

with open('kereta.csv') as fp :

table = from\_csv(fp)

print(table)

lanjut = True

while(lanjut) :

nomor = int(input('Nomor Kereta : '))

cekNomor = any(df['Nomor Kereta'] == nomor)

if cekNomor == True :

namaPemesan = str(input('Nama Pemesan : '))

jumlahTiket = int(input('Jumlah Tiket : '))

namaKereta = df.loc[df['Nomor Kereta'] == nomor]['Nama Kereta'].iloc[0]

kelas = df.loc[df['Nomor Kereta'] == nomor]['Kelas'].iloc[0]

kotaAsal = df.loc[df['Nomor Kereta'] == nomor]['Kota Asal'].iloc[0]

jamBerangkat = df.loc[df['Nomor Kereta'] == nomor]['Jam Berangkat'].iloc[0]

jamTiba = df.loc[df['Nomor Kereta'] == nomor]['Jam Tiba'].iloc[0]

harga = int(df.loc[df['Nomor Kereta'] == nomor]['Harga'].iloc[0])

statusBayar = ('Belum Dibayar')

tanggalBayar = ('0000-00-00 00:00:00')

querry = 'INSERT INTO datapemesan (NomorKereta, NamaPemesan,  NamaKereta, Kelas, KotaAsal, KotaTujuan, JamBerangkat, JamTiba, JumlahTiket, Harga, StatusBayar, TanggalPembayaran) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)'

data = (nomor, namaPemesan, namaKereta, kelas, kotaAsal, kotaTujuan, jamBerangkat, jamTiba, jumlahTiket, harga, statusBayar, tanggalBayar)

connect.execute(querry,data)

db.commit()

print('Pemesanan Berhasil Dilakukan Silahkan Melanjutkan ke Menu Pembayaran')

else :

print('Nomor Kereta Api Tidak Tersedia, Harap Masukan Kembali')

break

os.system('pause')

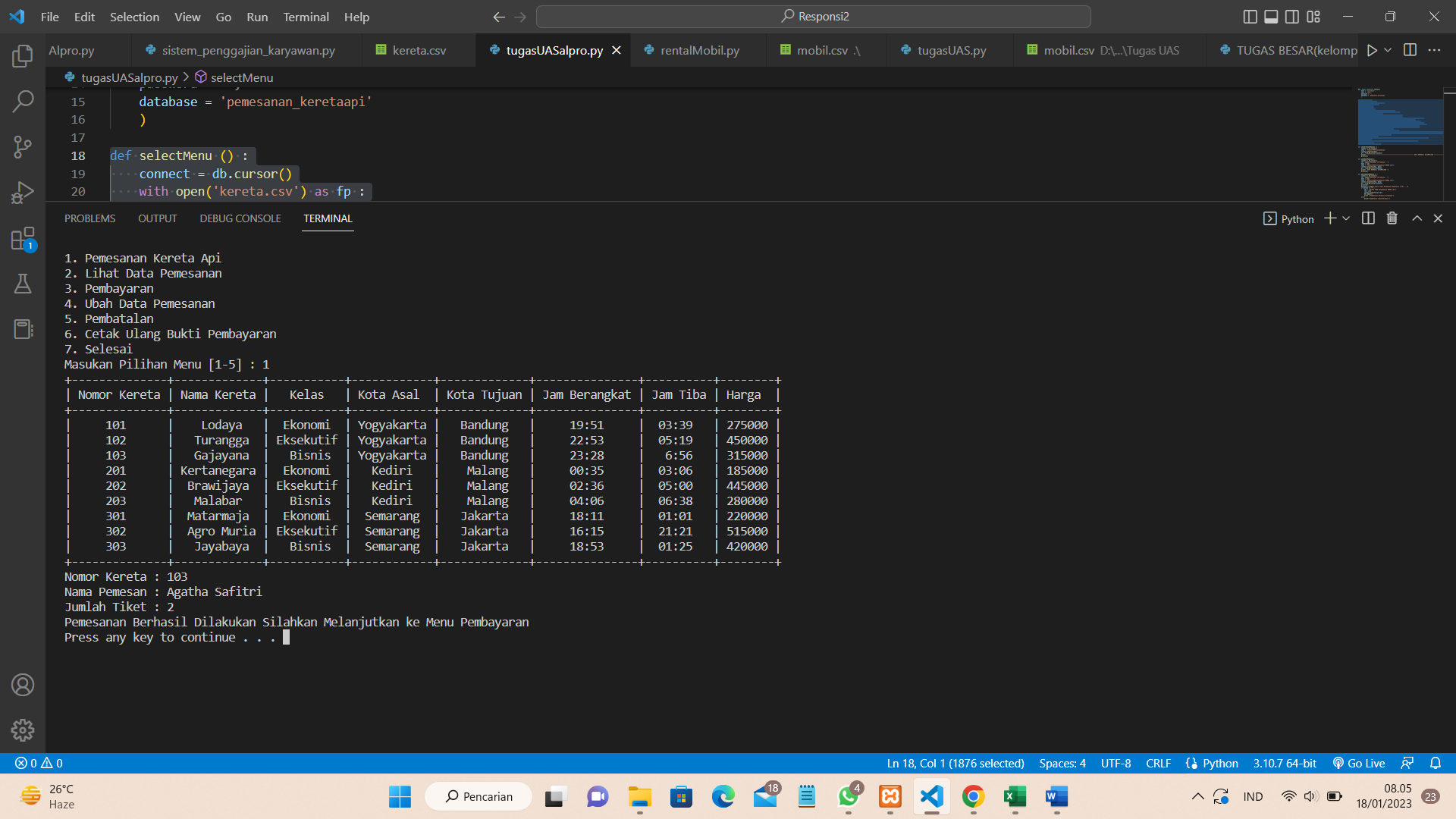
Dalam source code pemesanan kereta api menggunakan library pandas yang digunakan untuk mengimport data dari csv yang sudah dibuat. Selain pandas terdapat library pretty table yang berfungsi agar ketika pilihan menu daftar kereta ditampilkan ketika running tabel yang berisi informasi tersebut menjadi lebih rapi. Terdapat juga fungsi os yang fungsinya untuk membersihkan layar pada program yang akan dijalankan sehingga pemrogram dapat menenentukan saat untuk menghapus data yang telah dijalankan tanpa harus menutup program tersebut dan membukanya kembali.

Pada menu ini terdapat fungsi perulangan while apabila sebuah kondisi terpenuhi maka program akan terus melakukan perulangan, namun apabila suatu kondisi tidak terpenuhi maka perulangan akan berhenti.terdapat juga fungsi perulangan if else apabila data sudah sesuai maka akan langsung dieksekusi namun apabila data yang dimasukan salah maka perulangan akan dihentikan. Sedangkan sintak db.commit() digunakan untuk mengonfirmasi perubahan yang dibuat oleh pengguna ke database. Setiap kali ada perubahan yang dilakukan pada database perubahan tersebut perlu dilakukan. Jika kita tidak menggunakan metode commit() setelah melakukan perubahan maka database tidak akan diperbarui.

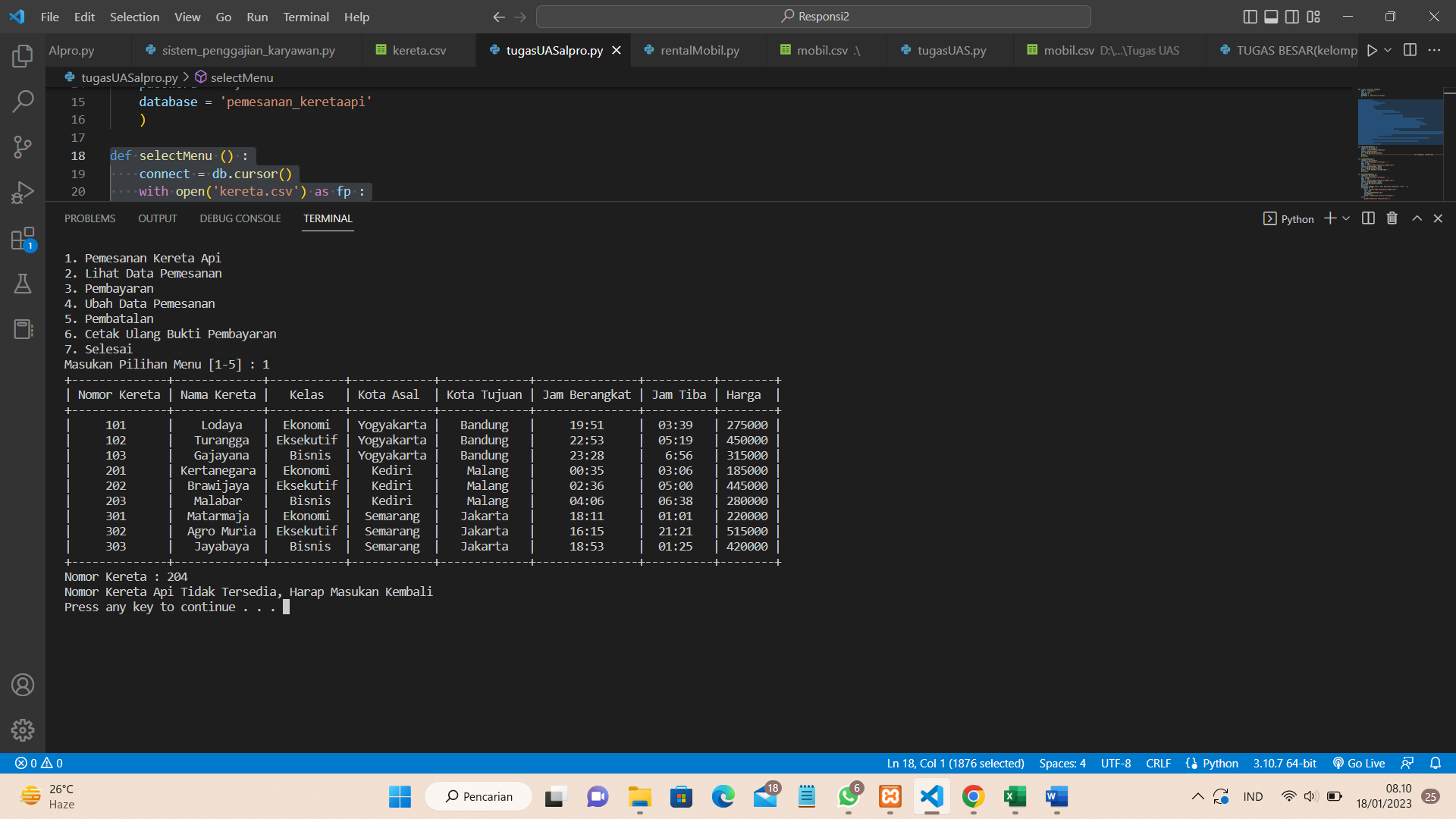
Sebuah gambar berisi teks, cuplikan layar, elektronik, komputer

Deskripsi dibuat secara otomatis

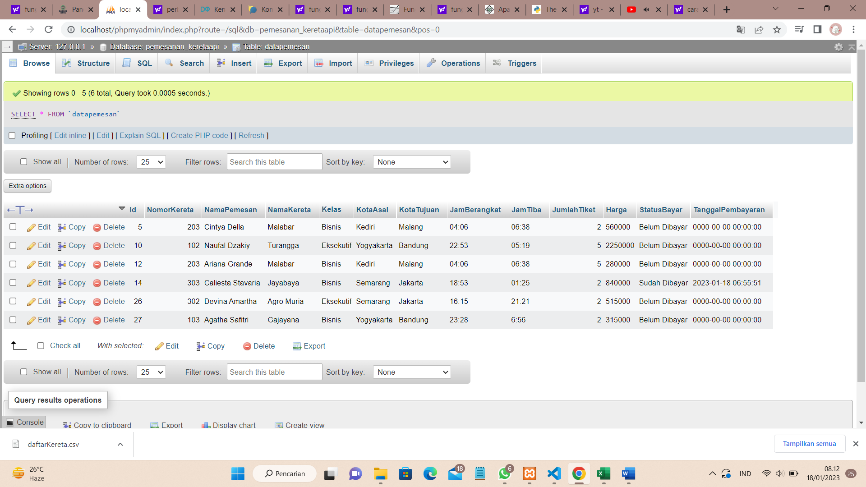
Gambar diatas merupakan file csv yang penulis gunakan untuk membuat Aplikasi Sederhana Sistem Rental Mobil.



Hasil running untuk melakukan pemesanan kereta api, pengguna harus menginputkan nomor kereta, nama pemesan dan jumlah tiket. Apabila berhasil maka akan muncul pernyataan “Pemesanan Berhasil Dilakukan. Silahkan Melanjutkan ke Menu Pembayaran”.



Apabila nomor kereta yang di masukan salah maka akan muncul pernyataan “Nomor Kereta Tidak Tersedia, Harap Masukan Kembali”.



Apabila sudah berhasil melakukan pemesanan data pemesan akan otomatis masuk ke database.

### **Menu Lihat Data Pemesan**

|  |  |
| --- | --- |
|  | Source code untuk melihat data pemesanan. Dalam source code untuk melihat data dari salah satu pemesan terdapat fungsi library pretty table yang gunanya untuk membuat tabel hasil running yang lebih terstruktur dan rapi. |

Sebuah gambar berisi teks, monitor, cuplikan layar, layar

Deskripsi dibuat secara otomatis

Hasil running menu ke 2 yaitu untuk melihat data pemesan, pengguna perlu menginputkan ID Pemesan agar dapat menampilkan data pemesanan.

### **12. Menu Pembayaran**

def pembayaranPesanan () :

    connect = db.cursor()

    connect.execute('SELECT\*FROM datapemesan')

    result = connect.fetchall()

    id = int(input('Masukkan ID Pemesan : '))

    for i in result:

        if id == i[0] :

            data = [id]

            show = 'SELECT\*FROM datapemesan WHERE id=%s'

            connect.execute(show, data)

            pt = from\_db\_cursor(connect)

            print(pt)

            konfirm = str (input('Apakah Data Sudah Benar (Y/N) : '))

            if konfirm == 'Y' :

                print(f'Jumlah yang Harus Dibayar : Rp. {i[10]}')

                uang = int(input('Masukkan Uang Pembayaran  : Rp. '))

                hasil = uang - int(i[10])

                if uang >= int(i[10]) :

                    status = ('Sudah Dibayar')

                    querry = f'UPDATE datapemesan SET StatusBayar="{status}" WHERE id="{id}"'

                    connect.execute(querry)

                    db.commit()

                    dt = datetime.datetime.now()

                    print("Tanggal : ",dt.strftime("%d %m %Y,pukul%H:%M"))

                    querry = f'UPDATE datapemesan SET TanggalPembayaran="{dt}" WHERE id="{id}"'

                    connect.execute(querry)

                    db.commit()

                    print('------------ BUKTI PEMBAYARAN ------------')

                    print(     f'ID Pemesan           :   {i[0]}     ')

                    print(     f'Nama Pemesan         :   {i[2]}     ')

                    print(     f'Nomor Kereta         :   {i[1]}     ')

                    print(     f'Nama Kereta          :   {i[3]}     ')

                    print(     f'Kelas                :   {i[4]}     ')

                    print(     f'Kota Asal            :   {i[5]}     ')

                    print(     f'Kota Tujuan          :   {i[6]}     ')

                    print(     f'Jam Berangkat        :   {i[7]}     ')

                    print(     f'Jam Tiba             :   {i[8]}     ')

                    print(     f'Jumlah Tiket         :   {i[9]}     ')

                    print(     f'Harga                :   Rp. {i[10]}')

                    print('Tunai                :   Rp.',uang        )

                    print('Kembalian            :   Rp.',hasil       )

                    print(     f'Tanggal Pembayaran   :   {dt}    ')

                    print('----- PEMBAYARAN BERHASIL DILAKUKAN ------')

                elif uang < int(i[10]) :

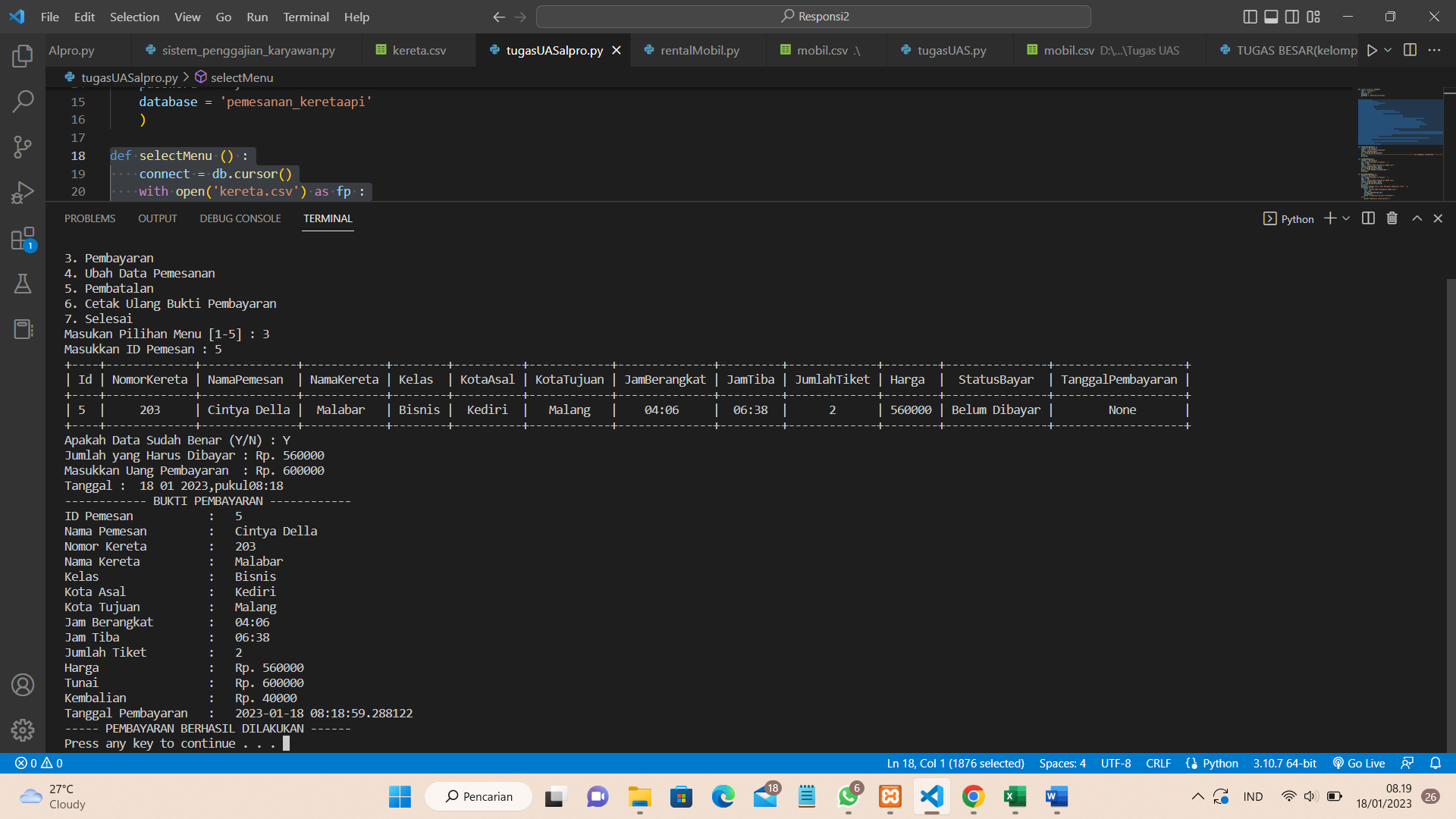
                    print ('Uang Pembayaran Kurang')

            else:

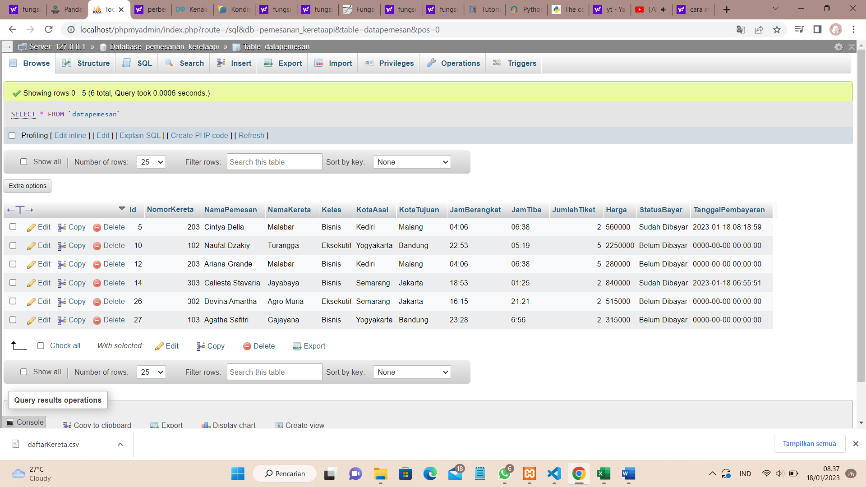
                break

Dalam source code pembayaran menggunakan library pretty table table yang berfungsi agar ketika data pemesan ditampilkan ketika running tabel yang berisi informasi tersebut menjadi lebih rapi. Terdapat juga library datetime yang berfungsi untuk menampilkan pada tanggal berapa pembayaran tersebut dilakukan.

Selain library terdapat fungsi perulangan for, dalam syntax for i in result arti subject “i” adalah item yang diekstrak dari result. Result tersebut diambil berdassarkan data yang terdapat dalam databse. Terdapat juga fungsi perulangan if else yang dimana ketika data yang diinputkan memenuhi kondisi pertama maka akan langsung dieksekusi tetapi jika data yang diinputkan tidak memenuhi kondisi pertama akan dialihkan ke kondisi kedua jika data sesuai maka akan dieksekusi. Namun apabila data tidak sesuai akan dialihkan ke kondisi selanjutnya jika nilai yang diinputkan tidak memenuhi semua kondisi maka perulangan akan dihentikan.



Hasil running dari source code pembayaran, pengguna harus menginputkan ID Pemesanan terlebih dahulu. Kemudian data pemesan akan ditampilkan dalam bentuk tabel dan akan muncul validasi apakah data sudah benar? Jika data salah maka pembayaran akan dibatalkan dan pengguna bisa mengubah data pada pilihan menu ke 4. Namun apabila data sudah benar maka akan ditampilkan jumlah yang harus dibayar dan pengguna harus menginputkan uang pembayaran jika uang yang dibayarkan sama dengan jumlah yang harus dibayar atau lebih dari jumlah yang harus dibayar maka akan dicetak bukti pembayaran, tetapi jika uang pembayaran kurang dari total pembayaran maka pembayaran akan otomatis dibatalkan oleh sistem.



Jika pembayaran sudah berhasil dilakukan maka data dalam database akan berubah status bayar yang awalnya belum dibayar akan berubah menjadi sudah dibayar, dan pada tanggal pembayaran akan berubah waktu dimana pengguna melakukan pembayaran.

### **13. Menu Ubah Data Pemesan**

def updateDataPemesan ():

    connect = db.cursor()

    id = input('Masukkan ID Pemesan : ')

    data = [id]

    show = 'SELECT\*FROM datapemesan WHERE id=%s'

    connect.execute(show, data)

    pt = from\_db\_cursor(connect)

    print(' DATA BERHASIL DITAMPILKAN ')

    print(pt)

    konfirm = str (input('Apakah Anda ingin Mengubah Pesanan (Y/N) : '))

    if konfirm == 'Y' :

        with open('kereta.csv') as fp :

            table = from\_csv(fp)

            print(table)

            lanjut = True

            while(lanjut) :

                nomor = int(input('Nomor Kereta : '))

                cekNomor = any(df['Nomor Kereta'] == nomor)

                if cekNomor == True :

                    jumlahTiket = int(input('Jumlah Tiket : '))

                    namaKereta = df.loc[df['Nomor Kereta'] == nomor]['Nama Kereta'].iloc[0]

                    kelas = df.loc[df['Nomor Kereta'] == nomor]['Kelas'].iloc[0]

                    kotaAsal = df.loc[df['Nomor Kereta'] == nomor]['Kota Asal'].iloc[0]

                    kotaTujuan = df.loc[df['Nomor Kereta'] == nomor]['Kota Tujuan'].iloc[0]

                    jamBerangkat = df.loc[df['Nomor Kereta'] == nomor]['Jam Berangkat'].iloc[0]

                    jamTiba = df.loc[df['Nomor Kereta'] == nomor]['Jam Tiba'].iloc[0]

                    harga = int(df.loc[df['Nomor Kereta'] == nomor]['Harga'].iloc[0])

                    statusBayar = ('Belum Dibayar')

                    tanggalBayar = ('0000-00-00 00:00:00')

                    querry = f'UPDATE datapemesan SET NomorKereta="{nomor}", NamaKereta="{namaKereta}", Kelas="{kelas}", KotaAsal="{kotaAsal}", KotaTujuan="{kotaTujuan}", JamBerangkat="{jamBerangkat}", JamTiba="{jamTiba}", JumlahTiket="{jumlahTiket}", Harga="{harga}", StatusBayar="{statusBayar}", TanggalPembayaran="{tanggalBayar}" WHERE id="{id}"'

                    connect.execute(querry)

                    show = 'SELECT\*FROM datapemesan WHERE id=%s'

                    connect.execute(show, data)

                    pt = from\_db\_cursor(connect)

                    print(pt)

                    db.commit()

                    print('Data Pemesanan Berhasil Diubah')

                    break

                else :

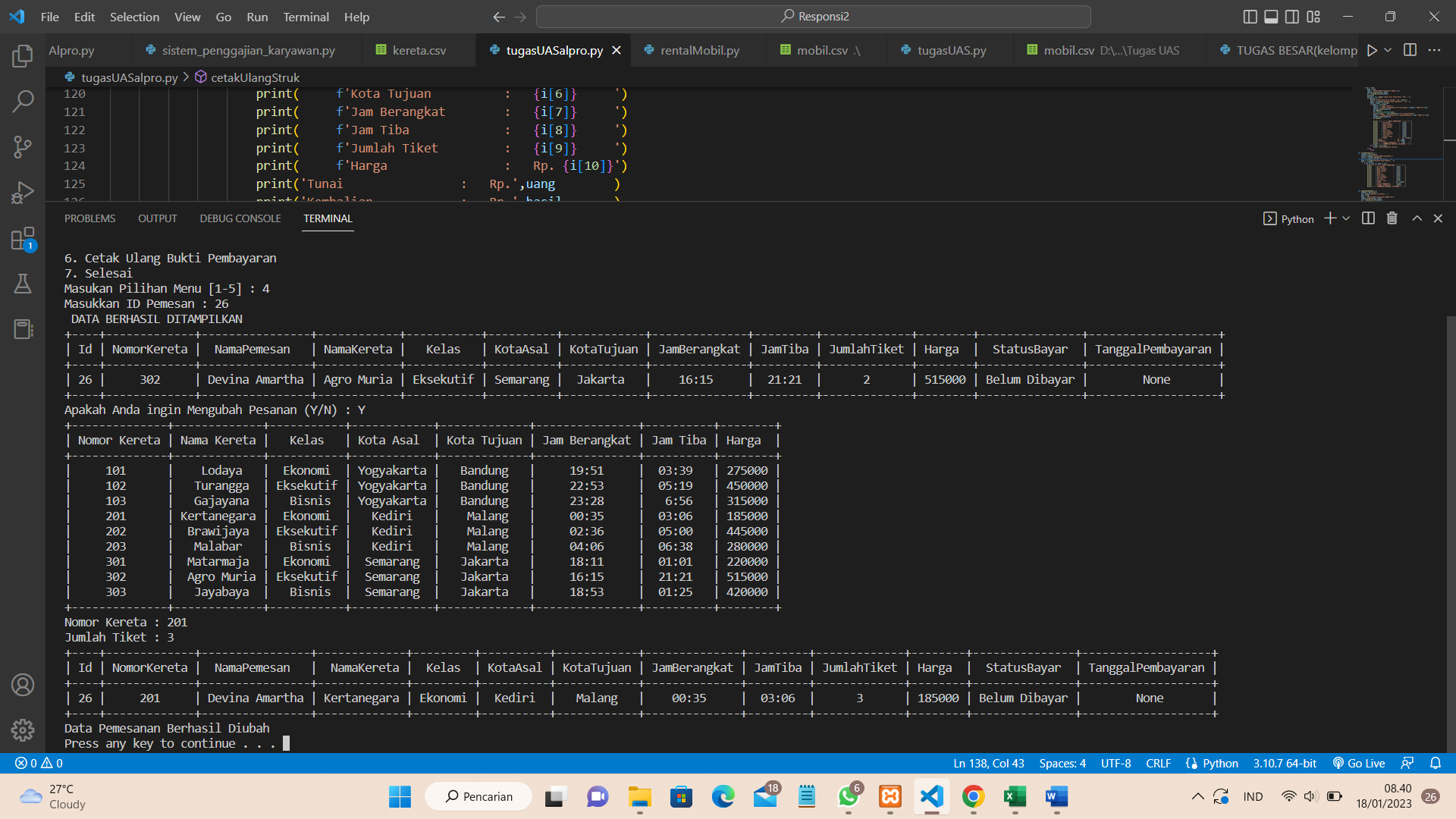
                    print('Nomor Kereta yang Anda Masukan Salah')

                    break

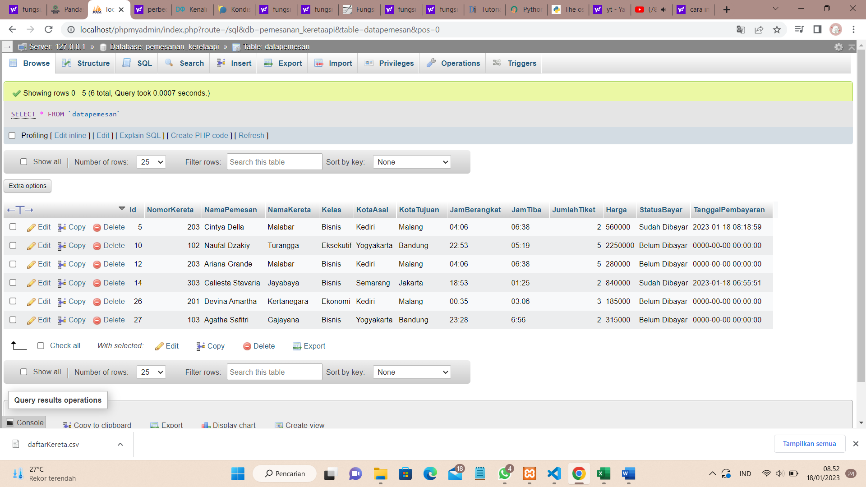
                os.system('pause')

Dalam source code pemesanan kereta api menggunakan library pretty table yang berfungsi agar ketika data pemesan ditampilkan ketika running tabel yang berisi informasi tersebut menjadi lebih rapi. Terdapat juga fungsi os yang fungsinya untuk membersihkan layar pada program yang akan dijalankan sehingga pemrogram dapat menenentukan saat untuk menghapus data yang telah dijalankan tanpa harus menutup program tersebut dan membukanya kembali.

Pada menu ini terdapat fungsi perulangan while apabila sebuah kondisi terpenuhi maka program akan terus melakukan perulangan, namun apabila suatu kondisi tidak terpenuhi maka perulangan akan berhenti. Terdapat juga fungsi perulangan if else apabila data sudah sesuai maka akan langsung dieksekusi namun apabila data yang dimasukan dalah maka perulangan akan dihentikan. Sedangkan sintak db.commit() digunakan untuk mengonfirmasi perubahan yang dibuat oleh pengguna ke database. Setiap kali ada perubahan yang dilakukan pada database perubahan tersebut perlu dilakukan. Jika kita tidak menggunakan metode commit() setelah melakukan perubahan maka database tidak akan diperbarui.

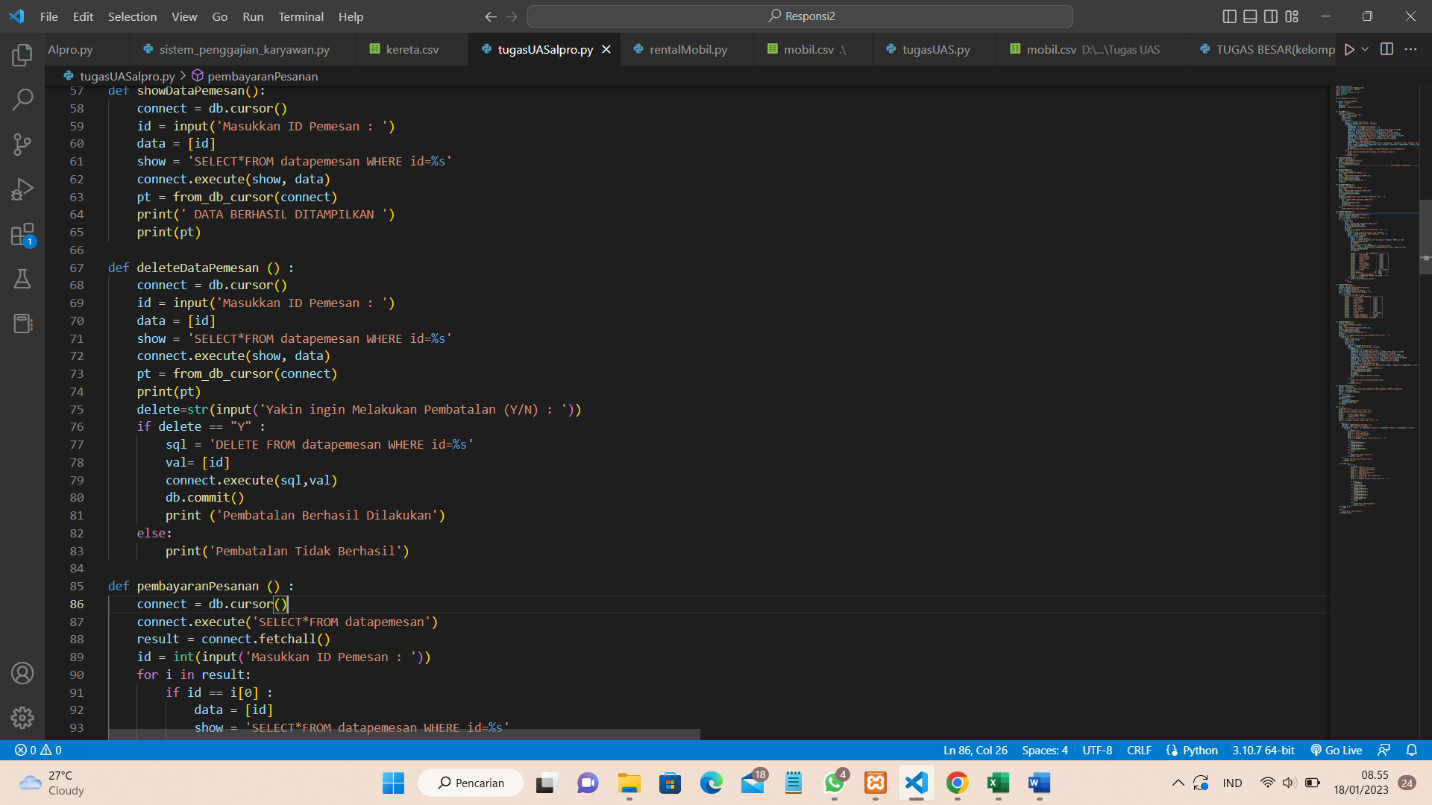


Hasil running dari source code update data, pengguna harus menginputkan ID Pemesan terlebih dahulu. Kemudian data pemesan akan ditampilkan dalam bentuk tabel dan akan muncul validasi apakah Anda ingin melakukan perubahan pesananan? Apabila tidak maka proses perubahan data pemesanan akan dihentikan. Namun apabila iya maka pengguna harus menginputkan nomor dan jumlah kereta setelah penginputan selesai maka akan muncul tabel baru yang berisi data pengguna yang sudah diperbarui dan muncul pernyataan “Data Pemesan Berhasil Diubah”. Data juga otomatis akan masuk ke database.

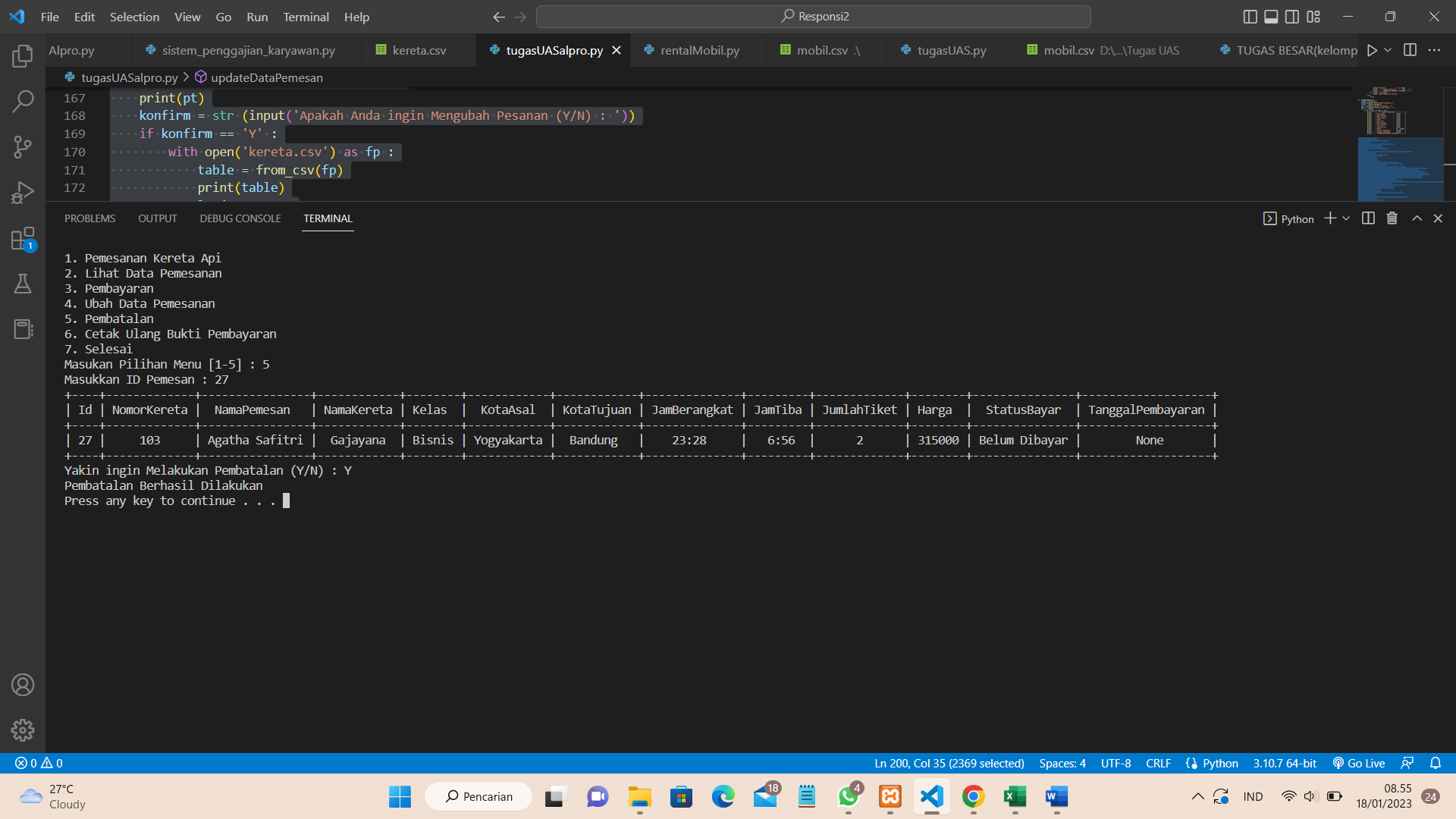


Jika dilihat di database pada ID Pemesan yang ke “26” data yang ditampilkan sudah berhasil di ubah.

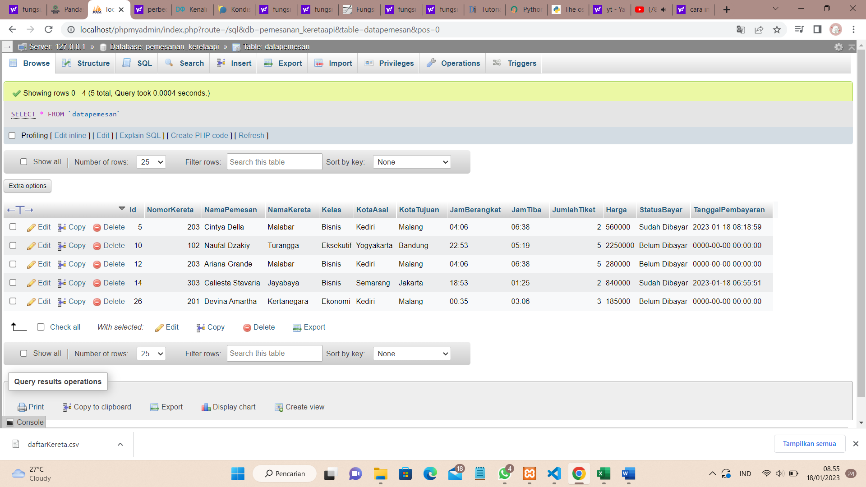
### **14. Menu Pembatalan**



Terdapat library prettytable dalam menu pembatalan yang berfungsi untuk membuat tampilan data lebih rapi dan terstruktur. Terdapat juga fungsi perulangan if else apabila kondisi terpenuhi akan dieksekusi dan apabila data yang iinputkan tidak sesuai dengan kondisi if maka akan dialihkan ke else ddan proses perulangan dihentikan.



Dalam hasil running pengguna harus menginputkan ID Pemesan kemudian sistem akan menanpilkan data dalam bentuk tabel dan akan muncul validasi yakin ingin melakukan pembatalan? Apabila tidak maka pembatalan tidak berhasil dan apabila iya maka akan muncul pernyataan “Pembatalan Berhasil Dilakukan”.



Dalam database ID pengguna “27” sudah tidak ada yang berarti proses pembatalan telah berhasil dilakukan.

### 15. Menu Cetak Ulang Bukti Pembayaran

|  |  |
| --- | --- |
|  | Terdapat fungsi perulangan for, dalam syntax for i in result arti subject “i” adalah item yang diekstrak dari result. Result tersebut diambil berdasarkan data yang terdapat dalam databse. Terdapat juga fungsi perulangan if yang dimana ketika data yang diinputkan memenuhi kondisi pertama maka akan langsung dieksekusi tetapi jika data yang diinputkan tidak memenuhi kondisi pertama akan dialihkan ke kondisi kedua jika data sesuai maka akan dieksekusi. Namun apabila data tidak sesuai akan dialihkan ke kondisi selanjutnya jika nilai yang diinputkan tidak memenuhi semua kondisi maka perulangan akan dihentikan. |

|  |  |
| --- | --- |
|  | Hasil running cetak ulang bukti pembayaran, pengguna harus menginputkan ID dan nama pemesan. Apabila terdapat salah satu atau kedua data yang diinputkan salah maka cetak ulang bukti pembayaran tidak berhasil dilakukan, namun apabila kedua data yang dimasukan benar maka bukti pembayaran akan dicetak. |

# **BAB IV**

# **PENUTUP**

## **KESIMPULAN**

Aplikasi Sederhana Pemesanan Kereta Api akan sangat membantu pemesan tiket atau pelanggan. karena selain fitur pemesanan aplikasi ini juga terdapat fitur ubah data pemesan yang dapat digunakan oleh pengguna apabila ingin melakukan perubahan data dan juga terdapat fitur pembatalan yang dapat digunakan pengguna apabila sewaktu-waktu ingin membatalkan pemesanan tiket kereta api.

## **SARAN**

Saran penulis dalam perancangan Aplikasi Sederhana Pemesanan Kereta Api adalah agar dibuatkan fitur untuk memilih nomor kursi agar ketika sudah melakukan pemesanan pelanggan tidak perlu kembali melakukan konfirmasi ke admin.

# **DAFTAR PUSTAKA**

<https://www.academia.edu/8443689/BAB_I_sistem_pemesanan_tiket_kereta_api>

<https://basangdata.com/pandas-menampilkan-data/>