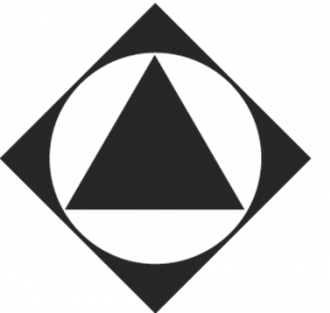
**TUGAS AKHIR**

**PROGRAM PYTHON**

**BOOKING HOTEL : HOTELA**

Disusun untuk memenuhi Nilai Tugas Akhir mata kuliah IFB-207 Praktikum Pemrograman Dasar



Ditulis oleh :

|  |  |
| --- | --- |
| Nama | : Silvy Nur Azkia |
| NIM | : 15-2022-178 |
| Kelas | : EE |

Prodi Informatika

Fakultas Teknologi Industri

Institut Teknologi Nasional

Bandung

2023

**KATA PENGANTAR**

Puji syukur kehadirat Tuhan Yang Maha Kuasa atas segala limpahan rahmat, taufik, dan hidayah-Nya sehingga dapat menyelesaikan laporan untuk memnuhi Tugas Akhir mata kuliah IFB-207 Pemrograman Dasar (P) dengan judul : Booking Hotel : HOTELA.

Tugas akhir ini menjadi bentuk aplikasi sederhana untuk melakukan pemesanan kamar hotel secara online. Penerapan konsep struktur data, dan pengontrolan alur program, menjadi fokus utama dalam pengembangan aplikasi.

Menyampaikan ucapakan terima kasih kepada Kang Arkan dan juga rekan-rekan Asisten Lab Praktikum Pemrograman Dasar lainnya atas dukungan, arahan, serta wawasan yang sudah diberikan.

Adanya pemahaman terhadap keterbatasan dan potensi perbaikan dalam tugas akhir ini membuat setiap masukan dan saran dari pembaca menjadi sangat berharga untuk pengembangan lebih lanjut.

Bandung, 11 Januari 2024

Silvy Nur Azkia

**Daftar Isi**

[BAB I 1](#_Toc155909553)

[**1.1** **Latar Belakang** 1](#_Toc155909554)

[**1.2** **Rumusan Masalah** 1](#_Toc155909555)

[**1.3** **Tujuan** 1](#_Toc155909556)

[**1.4** **Manfaat** 2](#_Toc155909557)

[**1.5** **Sistematika Penulisan** 2](#_Toc155909558)

[BAB II 3](#_Toc155909559)

[LANDASAN TEORI 3](#_Toc155909560)

[**2.1** **Software yang diguanakan** 3](#_Toc155909561)

[**2.2** **Bahasa Pemrograman yang Digunakan** 4](#_Toc155909562)

[**2.3** **Kondisional (if-else)** 5](#_Toc155909563)

[**2.4** **Array** 5](#_Toc155909564)

[**2.5** **Looping for** 5](#_Toc155909565)

[**2.6** **Looping While True** 5](#_Toc155909566)

[**2.7** **Library yang digunakan** 5](#_Toc155909567)

[**BAB III** 6](#_Toc155909568)

[**3.1** **Fitur pada aplikasi** 6](#_Toc155909569)

[**4.2.1** **Halaman login** 6](#_Toc155909570)

[**4.2.2** **Halaman Home Admin** 6](#_Toc155909571)

[**4.2.3** **Halaman Home User** 6](#_Toc155909572)

[**4.2.4** **Pilihan Kamar** 6](#_Toc155909573)

[**4.2.5** **Booking Kamar** 6](#_Toc155909574)

[**4.2.6** **Riwayat Transaksi.** 6](#_Toc155909575)

[**4.2.7** **Grafik Penjualan** 7](#_Toc155909576)

[**4.2.8** **Manajemen Data Kamar** 7](#_Toc155909577)

[**4.2.9** **Logout** 7](#_Toc155909578)

[**3.2** **Flowchart** 8](#_Toc155909579)

[**3.2.1** **Function select** 8](#_Toc155909580)

[**3.2.2** **Function insert** 9](#_Toc155909581)

[**3.2.3** **Funtion delete** 10](#_Toc155909582)

[**3.2.4** **Funtion register** 11](#_Toc155909583)

[**3.2.5** **Function validasi password** 11](#_Toc155909584)

[**3.2.6** **Function validasi password** 11](#_Toc155909585)

[**3.2.7** **Function login** 12](#_Toc155909586)

[**3.2.8** **Function Select Room** 13](#_Toc155909587)

[**3.2.9** **Function tambah data kamar** 14](#_Toc155909588)

[**3.2.10** **Functin update data** 15](#_Toc155909589)

[**3.2.11** **Function delete kamar** 16](#_Toc155909590)

[**3.2.12** **Function untuk menambahkan data ke riwayat** 16](#_Toc155909591)

[**3.2.13** **Function riwayat pembelian** 17](#_Toc155909592)

[**3.2.14** **Funtion riwayat pembelian user** 18](#_Toc155909593)

[**3.2.15** **Function untuk menampilkan pilihan kamar** 19](#_Toc155909594)

[**3.2.16** **Function menampilkan grafik penjualan** 20](#_Toc155909595)

[**3.2.17** **Function untuk booking kamar** 21](#_Toc155909596)

[**3.2.18** **Lanjutan function untuk booking kamar** 22](#_Toc155909597)

[**3.2.19** **Funtion untuk menu login** 23](#_Toc155909598)

[**3.2.20** **Function untuk main program** 24](#_Toc155909599)

[**3.2.21** **Lanjutan function untuk main program** 25](#_Toc155909600)

[**3.2.22** **Function untuk menambahkan user tidap kali registrasi** 26](#_Toc155909601)

[**3.2.23** **Program utama** 26](#_Toc155909602)

[BAB IV 27](#_Toc155909603)

[**4.1** **Source code** 27](#_Toc155909604)

[**4.1.1** **Library dan bagian atas pada kode** 27](#_Toc155909605)

[**4.1.2** **Code untuk perintah tampilkan** 27](#_Toc155909606)

[**4.1.3** **Code untuk menambahkan data pada database** 27](#_Toc155909607)

[**4.1.4** **Code untuk mengupdate data tertentu pada database** 27](#_Toc155909608)

[**4.1.5** **Code untuk menghapus data dalam tabel database** 27](#_Toc155909609)

[**4.1.6** **Code untuk menambahkan riwayat booking** 28](#_Toc155909610)

[**4.1.7** **Code untuk menampilkan riwayat booking role admin** 28](#_Toc155909611)

[**4.1.8** **Code untuk menampilkan riwayat booking user** 28](#_Toc155909612)

[**4.1.9** **Code untuk menampilkan grafik penjualan role admin** 28](#_Toc155909613)

[**4.1.10** **Code untuk validasi password saat registrasi** 29](#_Toc155909614)

[**4.1.11** **Code untuk validasi password saat registrasi** 29](#_Toc155909615)

[**4.1.12** **Code untuk registrasi** 29](#_Toc155909616)

[**4.1.13** **Code untuk select kamar** 29](#_Toc155909617)

[**4.1.14** **Code untuk membuat row baru di tabel user jika ada yang registrasi** 29](#_Toc155909618)

[**4.1.15** **Code untuk booking kamar** 30](#_Toc155909619)

[**4.1.16** **Code untuk connect database** 30](#_Toc155909620)

[**4.1.17** **Code untuk close connection** 31](#_Toc155909621)

[**4.1.18** **Code untuk register user** 31](#_Toc155909622)

[**4.1.19** **Code untuk login** 31](#_Toc155909623)

[**4.1.20** **Code untuk select room** 31](#_Toc155909624)

[**4.1.21** **Code untuk menghapus kamar role admin** 31](#_Toc155909625)

[**4.1.22** **Code untuk update room role admin** 32](#_Toc155909626)

[**4.1.23** **Code untuk tambah data kamar role admin** 32](#_Toc155909627)

[**4.1.24** **Code untuk function main** 32](#_Toc155909628)

[**4.1.25** **Code untuk program utama** 34](#_Toc155909629)

[**4.2** **Screenshoot Hasil Program** 34](#_Toc155909630)

[BAB V 39](#_Toc155909631)

[PENUTUP 39](#_Toc155909632)

[KESIMPULAN 39](#_Toc155909633)

# BAB I

**PENDAHULUAN**

## **Latar Belakang**

Dalam era digital yang terus berkembang, sistem pemesanan hotel online telah menjadi inovasi penting dalam industri perhotelan. Transformasi ini bertujuan memberikan kemudahan akses bagi pelanggan dalam mendapatkan informasi tentang ketersediaan kamar, harga, dan fasilitas hotel di waktu dan tempat yang mereka inginkan. Selain itu, implementasi sistem ini juga berdampak positif pada efisiensi operasional pihak hotel, dengan mengurangi potensi kesalahan manusia, mempercepat proses, dan mengurangi biaya administratif.

Lebih dari sekadar melakukan transaksi, sistem pemesanan hotel online menciptakan pengalaman yang lebih memuaskan bagi pelanggan. Dengan menyediakan gambaran yang jelas mengenai kamar, dan fasilitas hotel, sistem ini memfasilitasi pengambilan keputusan yang lebih baik. Dalam pengembangan program ini, memanfaatkan berbagai fitur Python seperti penggunaan struktur data, fungsi, dan modularitas untuk menciptakan aplikasi yang mudah dimengerti.

## **Rumusan Masalah**

Dari latar belakang diatas maka dapat ditarik bebrapa rumusan masalah, yaitu:

1. Bagaimana cara membuat aplikasi untuk booking hotel online?
2. Bagaimana cara kerja booking hotel HOTELA?
3. Apa saja fitur dan fungsionalitas booking hotel?
4. Apa saja perintah yang digunakan untuk menghasilkan aplikasi booking hotel HOTELA?

## **Tujuan**

Tujuan dari pembuatan laporan ini, yaitu :

1. Mengetahui cara pembuatana aplikasi berbasis pahasa Python untuk Booking Hotel
2. Mengetahui cara kerja aplikasi Booking Hotel
3. Mengetahui macam-macam fitur dan fungsionalitas yang digunakan di Booking Hotel HOTELA
4. Mengetahui apa saja perintah yang digunakan untuk membuat aplikasi booking hotel HOTELA
5. Memenuhi tugas akhir matakuliah IFB-207 Pemrograman Dasar.

## **Manfaat**

Manfaat dari pembuatan laporan ini, yaitu :

1. Mempelajari cara pembuatan aplikasi menggunakan Python.
2. Mengetahui cara kerja Python.
3. Memahami bahasa pemrograman yang digunakan aplikasi, yaitu Python.
4. Mengetahui kegunaan library yang tersedia di Python.
5. Membuat aplikasi yang dapat membantu pekerjaan menjadi lebih mudah dan praktis.

## **Sistematika Penulisan**

Sistematika penulisan dalam laporan ini disusun sebagai berikut :

**BAB I PENDAHULUAN**

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan, manfaat, sistematika penulisan.

**BAB II LANDASAN TEORI**

Bab ini menjelasakan tentang software yang digunakan dan bahsan pemrograman yang digunakan yaitu Python. Juga menjelaskan tentang perintah yang digunakan.

**BAB III PERANCANGAN**

Bab ini menjelaskan mengenai fungsi dan fitur yang ada pada aplikasi Booking Hotel yang akan dibuat, serta flowchart dari program yang dibuat.

**BAB IV IMPLEMENTASI**

Bab ini berisi sourecode dan screenshoot dari aplikasi yang dibuat, kemudian dijelaskan apa saja yang terjadi pada sourcecode.

**BAB V PENUTUP**

Bab ini berisikan kesimpulan dari program dan laporna yang telah dibuat.

# BAB II

# LANDASAN TEORI

## **Software yang diguanakan**

PyCharm adalah Integrated Development Environment (IDE) untuk pengembangan aplikasi menggunakan bahasa pemrograman Python. PyCharm adalah software yang membantu developer untuk mengembangkan aplikasi dengan menggunakan bahasa pemrograman Python yang dilengkapi GUI. PyCharm hadir dengan sejumlah besar modul, paket, dan alat untuk mempercepat pengembangan Python sambil mengurangi upaya yang diperlukan untuk melakukan hal yang sama secara besar-besaran secara bersamaan.

Gambar 2.1.1 Logo software Pycharm

1. Kelebihan Pycharm :
2. menyediakan editor kode yang cerdas dengan fitur penyorotan sintaks, autocompletion, dan refactorings yang dapat meningkatkan produktivitas pengembang.
3. Tersedia untuk berbagai platform termasuk Windows, macOS, dan Linux.
4. PyCharm menyediakan dukungan terintegrasi untuk virtual environment, memungkinkan pengembang membuat dan mengelola lingkungan Python yang terisolasi.
5. Memiliki fungsi SQL editor dan dukungan database yang dapat diakses langsung dari IDE, memudahkan pengembangan aplikasi yang melibatkan database.
6. Kekurangan Pycharm :
7. Untuk proyek-proyek besar, waktu pembukaan awal proyek mungkin memakan waktu yang cukup lama.
8. PyCharm memiliki banyak fitur yang memerlukan pembelajaran awal, dan pengguna mungkin perlu waktu untuk menguasai semua fitur tersebut.
9. PyCharm dikenal memiliki konsumsi memori yang tinggi, terutama saat digunakan untuk proyek-proyek besar. Hal ini dapat mempengaruhi kinerja pada mesin dengan sumber daya terbatas.

## **Bahasa Pemrograman yang Digunakan**

Bahasa python memiliki beberapa sintaks yang umumnya ada pada bahasa pemrograman lainnya seperti input/output proses, struktur seleksi, struktur pengulangan, pernyataan fungsi (sub program), dan sebagainya. Sebagai bahasa pemrograman yang populer dan komprehensif, python menggabungkan kapabilitas, sintaksi kode yang jelas serta dilengkapi dengan pustaka standar yang memiliki fungsionalitas yang besar. Python sendiri termasuk ke dalam kategori bahasa pemrograman tingkat tinggi. Python juga dikenal dengan bahasa pemrograman yang mudah dipelajari, karena struktur sintaknya rapi dan mudah dipahami.

Gambar 2.2.1 Logo Bahasa Python

1. Kelebihan bahasa Python
2. Python didesain dengan sintaks yang bersifat mudah dibaca dan dimengerti, membuatnya cocok untuk pemula maupun pengembang berpengalaman.
3. Python memiliki library yang kuat seperti Pandas, NumPy, dan Matplotlib yang mendukung analisis data, ilmu data, dan visualisasi data.
4. Sintaksis bersifat ekspresif dan pemeliharaan kode relatif mudah, memungkinkan pengembang untuk mengelola dan memperbarui proyek dengan efisien.
5. Python telah digunakan dalam pengembangan perangkat lunak berskala besar dan diterapkan dalam berbagai proyek besar dan organisasi.
6. Kekurangan bahasa Python
7. Python dapat memakan banyak memori, terutama pada aplikasi yang membutuhkan sumber daya tinggi.
8. Untuk aplikasi yang membutuhkan kecepatan pemrosesan tinggi atau memiliki persyaratan kecepatan real-time yang ketat, Python mungkin bukan pilihan yang ideal.
9. Python adalah bahasa yang tidak bersifat statis dan tidak menerapkan tipe data yang ketat, yang dapat menyebabkan kesalahan saat runtime.

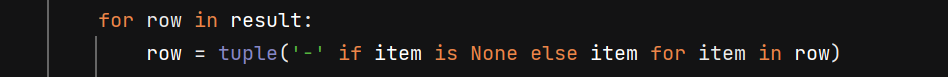
## **Kondisional (if-else)**

Kondisi yang digunakan yaitu if, elif, dan else yang digunakan untuk menangani pilihan pengguna, bagian else digunakan untuk menjalankan kode program apabila kondisi tidak terpenuhi (pengecualian).

## **Array**

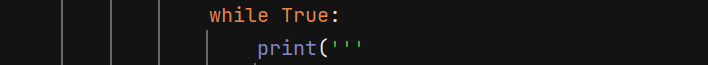
Ada penggunaan list Python yang dapat dianggap setara dengan array diantaranya yaitu list yang digunakan menyimpan dictionary untuk setiap riwayat transaksi.

## **Looping for**

Terdapat penggunaan loop for pada beberapa bagian kode untuk melakukan iterasi melalui hasil query atau item dalam suatu list untuk mengiterasi melalui hasil query

Gambar 2.5.1 Contoh penggunaan looping for

## **Looping While True**

Digunakan untuk membuat looping atau perulangan tidak terbatas selama kondisi masih bernilai ”True” atau benar, dapat dihentikan dengan pernyataan ”break”

Gambar 2.6.1 contoh penggunaan looping While True

## **Library yang digunakan**

Dalam aplikasi yangd ibuat memerlukan atau digunakan beberapa library, diantaranya :

1. ”mysql.connector” : Library ini digunakan untuk berinteraksi dengan database MySQL. Fungsinya mencakup koneksi ke database, eksekusi query, dan lainnya.
2. ”matplotlib.pylot” : Library ini digunakan untuk membuat visualisasi data, seperti untuk menampilkan grafik penjualan berdasarkan tipe kamar.
3. ”pandas” : Library ini digunakan untuk membantu dalam memproses hasil query database dan menampilkannya dalam bentuk grafik.
4. ”re” : Library ini adalah modul ekspresi reguler untuk melakukan pencocokan pola string, digunakan untuk memvalidasi username.
5. ”datetime” : Library ini digunakan untuk menampilkan tanggal saat melakukan pemesanan kamar.

### **BAB III**

**PERANCANGAN**

## **Fitur pada aplikasi**

Berikut merupakan fitur-fitur yang ada pada booking hotel, HOTELA:

### **Halaman login**

Pada halaman login terdapat fitur login, registrasi dan exit. Menggunakan looping While True yang break nya terletak apabila pengguna memilih exit, amak akan keluar dari aplikasi. Terdapat looping untuk memastikan apakah username dan password yang dimasukkan termasuk kedalam login admin atau sebagai user.

### **Halaman Home Admin**

Jika pada saat login termasuk ke dalam tabel login admin dalam databse, maka akan dimunculkan benyak pilihan perintah pada halaman home adminnya. Diantaranya bisa melakukan CRUD dan meanmpilkan grafik dari penjualan.

### **Halaman Home User**

Berbeda dengan halaman home admin, pada home user hanya terdapat beebrapa pilihan saja, diantarnya melihat kamar apa saja yang ada, lalu pengisian untuk booking, melihat riwayat booking, dan juga logout.

### **Pilihan Kamar**

Menampilkan daftar kamar beserta informasi tentang ketersediaan kamr, harga, type, dan lantai kamar berada.

### **Booking Kamar**

Bisa dilakukan melalui role admin ataupun user. Dengan menginputkan type kamar, no kamar, waktu check in dan waktu check out. Harga kamar akan muncul saat menginputkan berapa lama waktu inap yang dikali dengan harga dari type kamar tersebut.

### **Riwayat Transaksi.**

Jika pada role admin maka riwayat transaksi yang ditampilkan meliputi semua pengguna, sednagkan jika role user maka hanya akan menampilkan data dari booking yang telah dilakukannya, jika belum melkukan booking maka akan muncul notifikasi ”tidak ada data untuk ditampilkan”.

### **Grafik Penjualan**

Akan menampilkan grafik penjualan berdasarkan type kamar yang ada, menggunakan pandas dan amtplotlib untuk visualisasi data. Hanya bisa diakses oleh admin saja.

### **Manajemen Data Kamar**

Hanya tersedia untuk role admin, bisa menambahkan datakamr, mengupdate dataa kamar apabila status kamar tersebut berubah, dan dapat menghapus kamr.

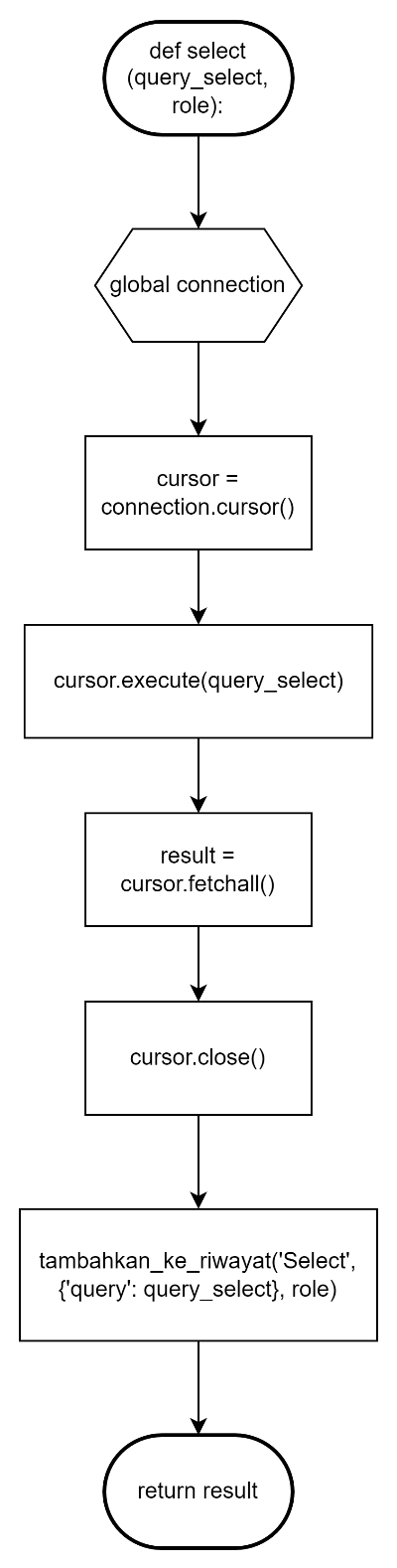
### **Logout**

Bisa diakses oleh kedua role, yang apabila memilih logout akan kembali diarahkan ke menu login untuk melakukan login ulang ataupun exit dari aplikasi.

## **Flowchart**

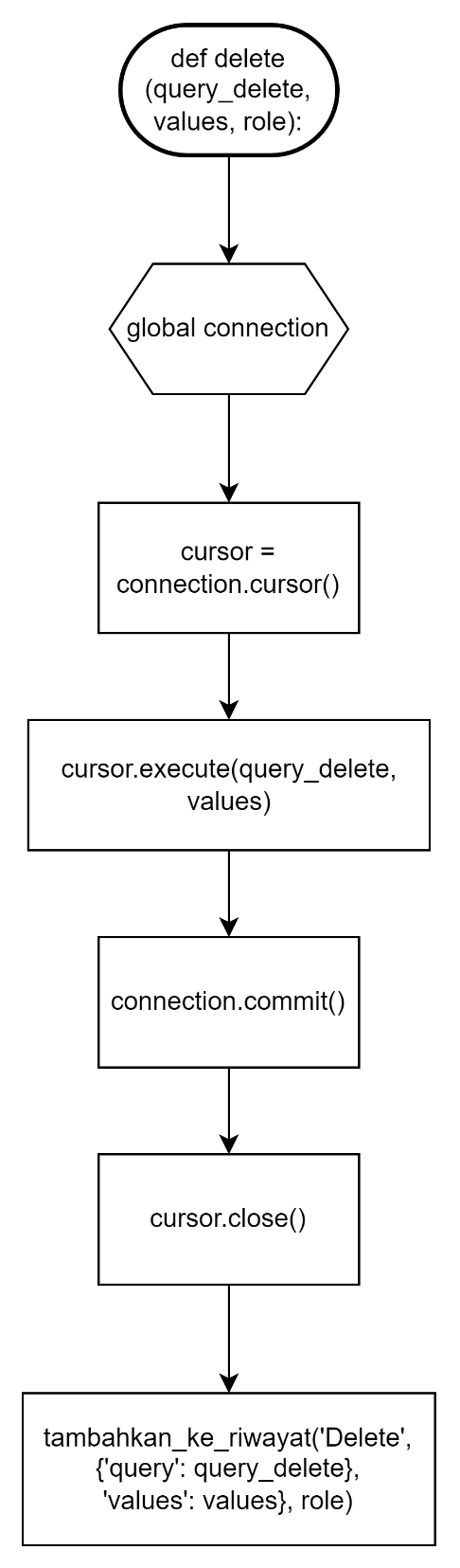
Berikut merupakan flowchart yang dirancang untuk aplikasi booking hotel, HOTELA

### **Function select**



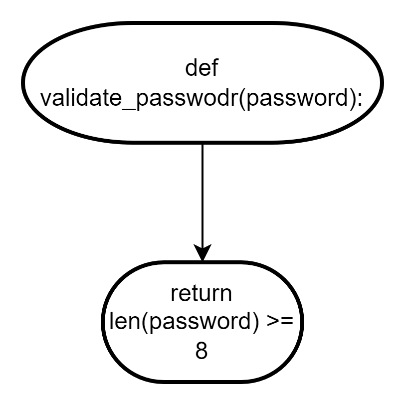
### **Function insert**

### **Funtion delete**



### **Funtion register**

### **Function validasi password**



### **Function validasi password**

### **Function login**

### **Function Select Room**

### **Function tambah data kamar**

### **Functin update data**



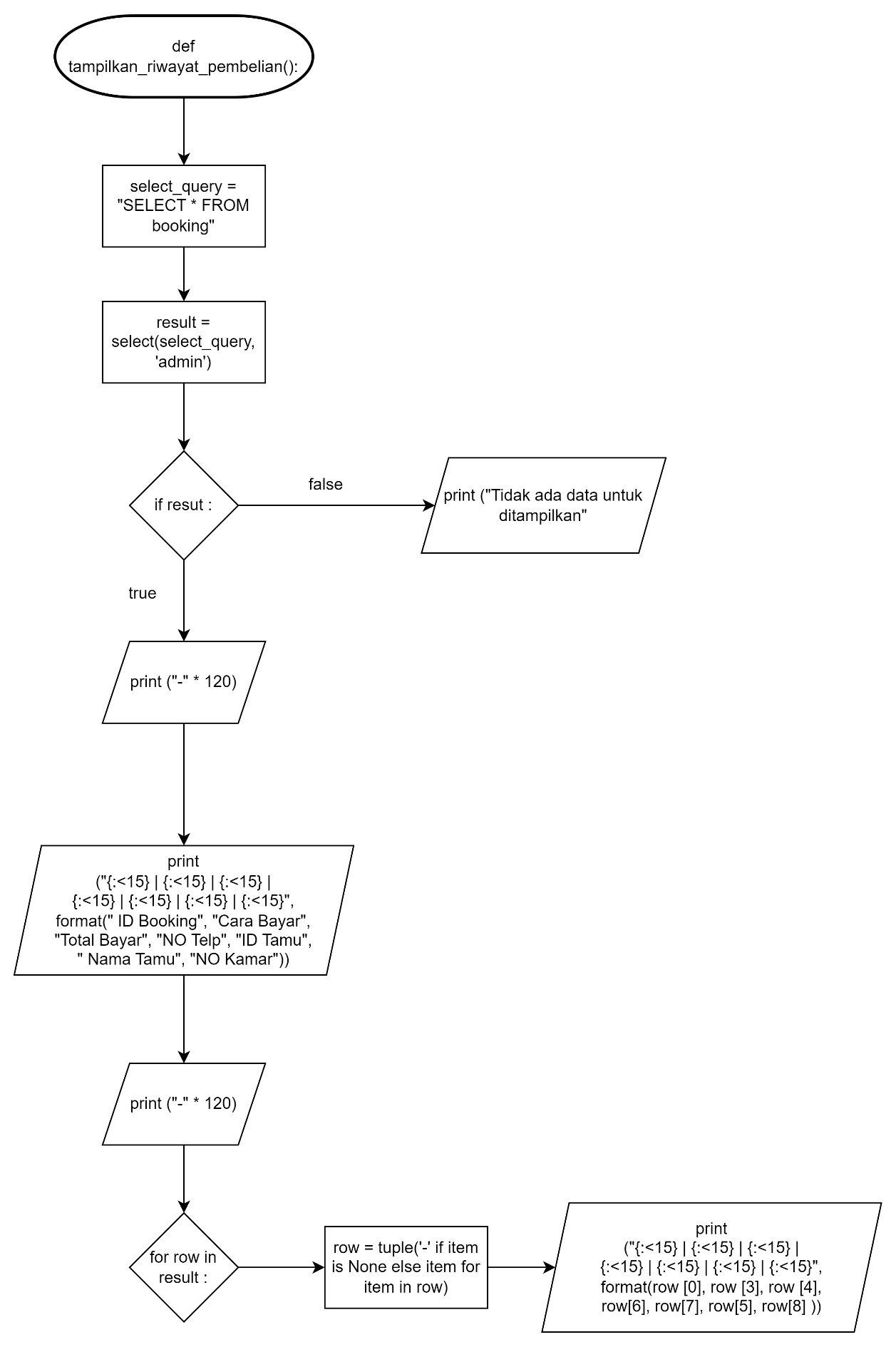
### **Function delete kamar**



### **Function untuk menambahkan data ke riwayat**



### **Function riwayat pembelian**

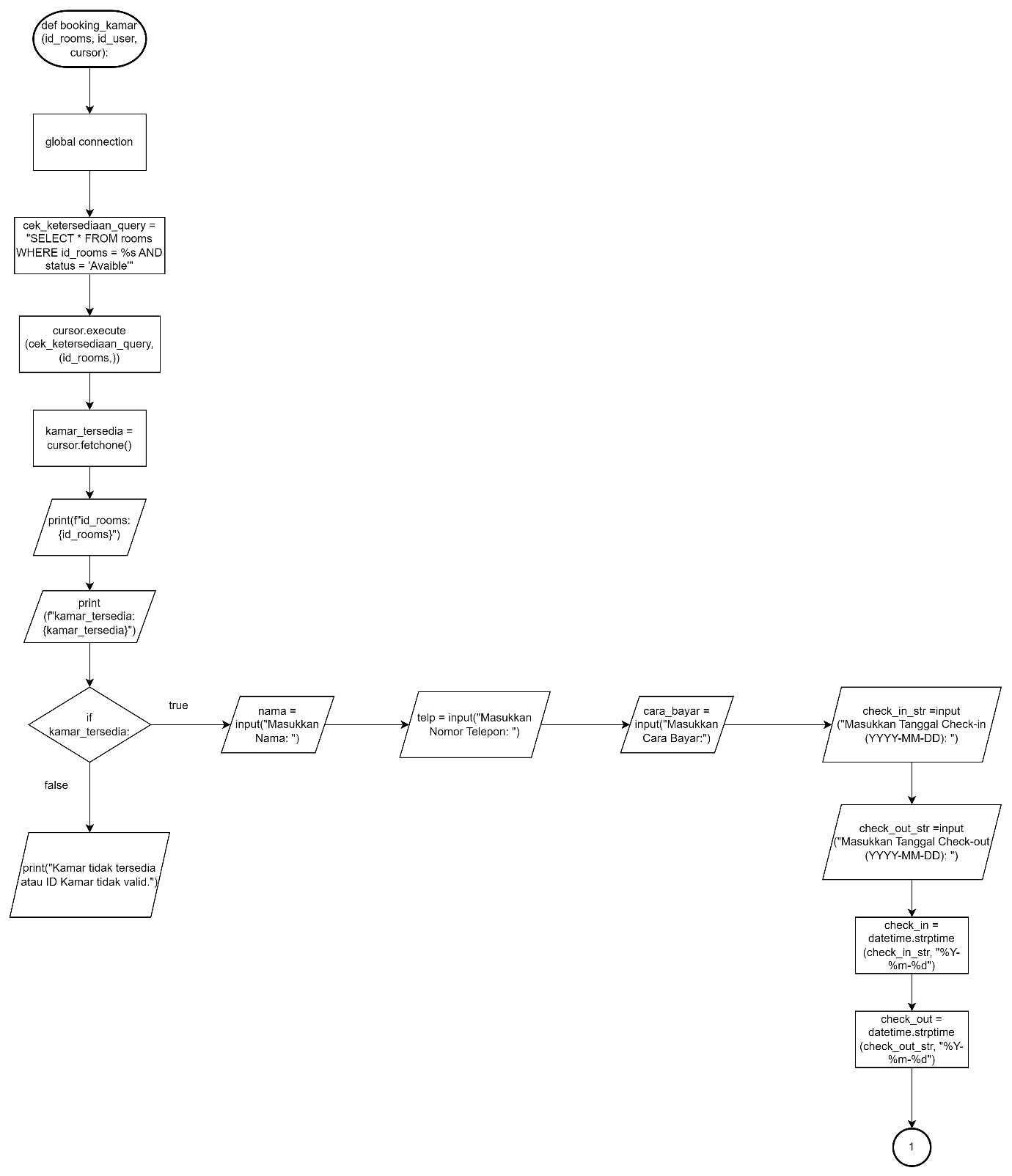


### **Funtion riwayat pembelian user**

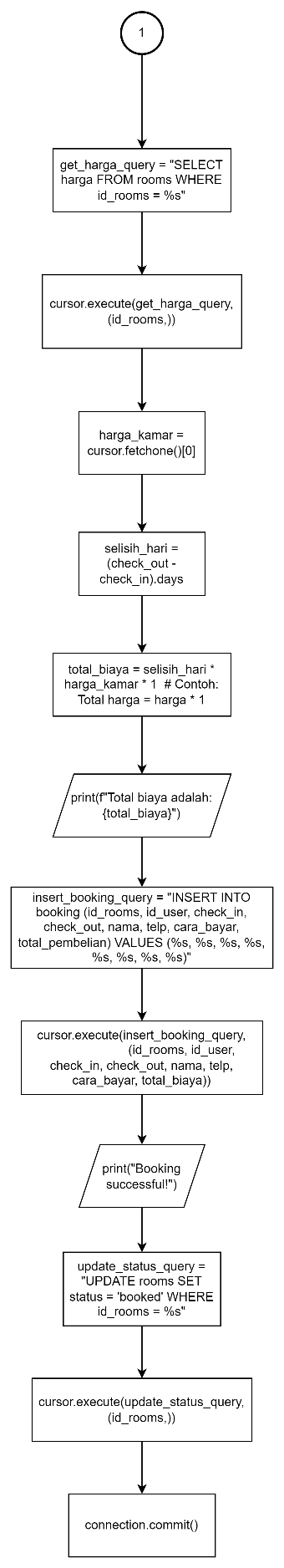
### **Function untuk menampilkan pilihan kamar**

### **Function menampilkan grafik penjualan**

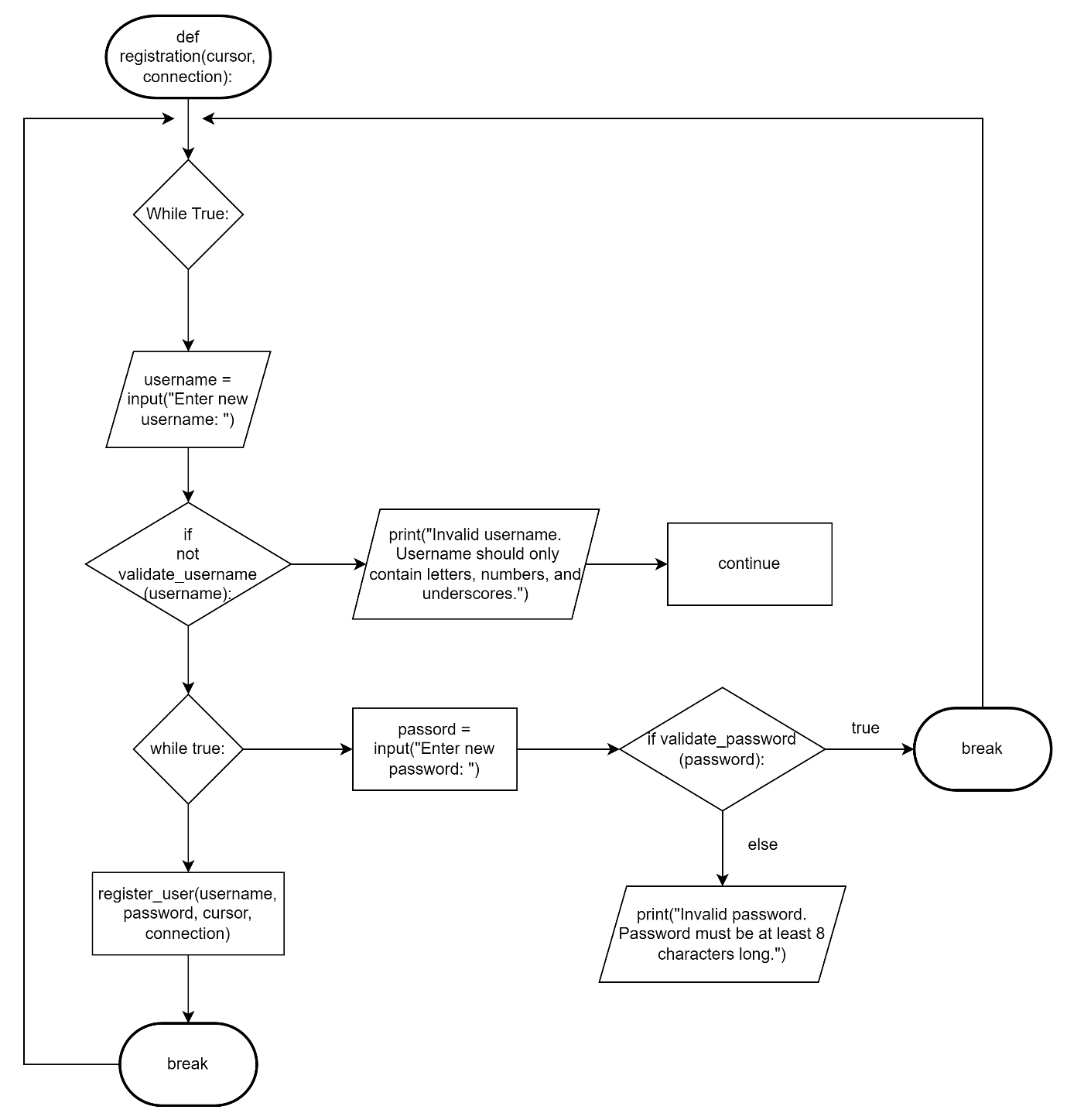
### **Function untuk booking kamar**



### **Lanjutan function untuk booking kamar**



### **Funtion untuk menu login**



### **Function untuk main program**

### **Lanjutan function untuk main program**

### **Function untuk menambahkan user tidap kali registrasi**



### **Program utama**

# BAB IV

**IMPLEMENTASI**

## **Source code**

Berikut merupakan source code dari program yang dibuat

### **Library dan bagian atas pada kode**

import mysql.connector

from mysql.connector import Error

import matplotlib.pyplot as plt

import pandas as pd

import re

from datetime import datetime

riwayat\_transaksi = [] #digunakan untuk menyimpan riwayat transaksi dalam bentuk list

connection: mysql.connector.MySQLConnection | None = None #koneksi global ke database MySQL

### **Code untuk perintah tampilkan**

def select(query\_select, role):  
 global connection cursor = connection.cursor()  
 cursor.execute(query\_select)  
 result = cursor.fetchall()  
 cursor.close()  
 tambahkan\_ke\_riwayat('Select', {'query': query\_select}, role) #untuk mencatat transaksi  
 return result

### **Code untuk menambahkan data pada database**

#untuk menambahkan data pada database

def insert(query\_insert, values, role):

global connection

cursor = connection.cursor()

cursor.execute(query\_insert, values)

connection.commit()

cursor.close()

tambahkan\_ke\_riwayat('Insert', {'query': query\_insert, 'values': values}, role)

### **Code untuk mengupdate data tertentu pada database**

#untuk mengupdate data tertentu pada database

def update(query\_update, new\_values, role):

global connection

cursor = connection.cursor()

cursor.execute(query\_update, new\_values)

connection.commit()

cursor.close()

tambahkan\_ke\_riwayat('Update', {'query': query\_update, 'new\_values': new\_values}, role)

### **Code untuk menghapus data dalam tabel database**

#untuk menghapus data dalam tabel database

def delete(query\_delete, values, role):

global connection

cursor = connection.cursor()

cursor.execute(query\_delete, values)

connection.commit()

cursor.close()

tambahkan\_ke\_riwayat('Delete', {'query': query\_delete, 'values': values}, role)

### **Code untuk menambahkan riwayat booking**

def tambahkan\_ke\_riwayat(action, detail, role):

riwayat\_transaksi.append({'action': action, 'detail': detail, 'role': role})

### **Code untuk menampilkan riwayat booking role admin**

def tampilkan\_riwayat\_pembelian():

select\_query = "SELECT \* FROM booking"

result = select(select\_query, 'admin') # Menampilkan riwayat dari tabel admin

if result:

print("-" \* 120)

print("{:<15} | {:<15} | {:<15} | {:<15} | {:<15} | {:<15} | {:<15}".format(

"ID Booking", "Cara Bayar", "Total Bayar", "NO Telp", "ID Tamu", "Nama Tamu", "NO Kamar"

))

print("-" \* 120)

for row in result:

row = tuple('-' if item is None else item for item in row)

print("{:<15} {:<15} {:<15} {:<15} {:<15} {:<15} {:<15}".format(

row[0], row[3], row[4], row[6], row[7], row[5], row[8]

))

else:

print("Tidak ada data untuk ditampilkan.")

### **Code untuk menampilkan riwayat booking user**

def riwayat\_pembelian\_user(id\_user, cursor):

select\_query = f"SELECT \* FROM booking where id\_user = {id\_user}"

result = select(select\_query, 'user')

if result:

print("-" \* 110)

print("{:<15} | {:<15} | {:<15} | {:<15} | {:<15} | {:<15}".format(

"ID Booking", "Cara Bayar", "Total Bayar", "NO Telp", "Nama Tamu", "NO Kamar"

))

print("-" \* 110)

for row in result:

row = tuple('-' if item is None else item for item in row)

print("{:<15} {:<15} {:<15} {:<15} {:<15} {:<15}".format(

row[0], row[3], row[4], row[6], row[5], row[8]

))

else:

print("Tidak ada data untuk ditampilkan.")

### **Code untuk menampilkan grafik penjualan role admin**

def tampilkan\_grafik\_penjualan():

select\_query = '''

SELECT rooms.type, COUNT(\*) as jumlah

FROM booking

INNER JOIN rooms ON booking.id\_rooms = rooms.id\_rooms

GROUP BY rooms.type

'''

result = select(select\_query, 'admin')

if result:

df = pd.DataFrame(result, columns=['type', 'jumlah'])

df.plot(kind='bar', x='type', y='jumlah', legend=False)

plt.title('Grafik Penjualan Berdasarkan Type Kamar')

plt.xlabel('Type Kamar')

plt.ylabel('Jumlah')

plt.show()

else:

print("Tidak ada data untuk ditampilkan.")

### **Code untuk validasi password saat registrasi**

def validate\_password(password):

# Minimal 8 karakter

return len(password) >= 8

### **Code untuk validasi password saat registrasi**

def validate\_username(username):

# Username hanya boleh mengandung huruf, angka, dan underscore

return re.match(r'^\w+$', username) is not None

### **Code untuk registrasi**

def registration(cursor, connection):

while True:

username = input("Enter new username: ")

if not validate\_username(username):

print("Invalid username. Username should only contain letters, numbers, and underscores.")

continue

while True:

password = input("Enter new password: ")

if validate\_password(password):

break

else:

print("Invalid password. Password must be at least 8 characters long.")

register\_user(username, password, cursor, connection)

break

### **Code untuk select kamar**

def tampilkan\_pilihan\_kamar(role):

rooms = select('SELECT \* FROM rooms', role)

if rooms:

print("-" \* 120)

print("{:<21} | {:<22} | {:<22} | {:<21} | {:<22}".format(

"ID Kamar", "Tipe", "Harga", "Lantai", "Ketersediaan"

))

print("-" \* 120)

for room in rooms:

room = tuple('-' if item is None else item for item in room)

print("{:<21} | {:<22} | {:<22} | {:<21} | {:<22}".format(\*room))

else:

print("Tidak ada data untuk ditampilkan.")

### **Code untuk membuat row baru di tabel user jika ada yang registrasi**

def create\_users\_table(cursor):

create\_table\_query = '''

CREATE TABLE IF NOT EXISTS users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL

)

'''

cursor.execute(create\_table\_query)

### **Code untuk booking kamar**

def booking\_kamar(id\_rooms, id\_user, cursor):

global connection # Tambahkan variabel connection global

# Cek apakah kamar tersedia berdasarkan ID kamar

cek\_ketersediaan\_query = "SELECT \* FROM rooms WHERE id\_rooms = %s AND status = 'Avaible'"

cursor.execute(cek\_ketersediaan\_query, (id\_rooms,))

kamar\_tersedia = cursor.fetchone()

print(f"id\_rooms: {id\_rooms}")

print(f"kamar\_tersedia: {kamar\_tersedia}")

if kamar\_tersedia:

# Meminta informasi tambahan dari pengguna

nama = input("Masukkan Nama: ")

telp = input("Masukkan Nomor Telepon: ")

cara\_bayar = input("Masukkan Cara Bayar:")

check\_in\_str = input("Masukkan Tanggal Check-in (YYYY-MM-DD): ")

check\_out\_str = input("Masukkan Tanggal Check-out (YYYY-MM-DD): ")

# Parsing tanggal menggunakan datetime

check\_in = datetime.strptime(check\_in\_str, "%Y-%m-%d")

check\_out = datetime.strptime(check\_out\_str, "%Y-%m-%d")

# Mendapatkan harga kamar

get\_harga\_query = "SELECT harga FROM rooms WHERE id\_rooms = %s"

cursor.execute(get\_harga\_query, (id\_rooms,))

harga\_kamar = cursor.fetchone()[0]

# Menghitung selisih hari

selisih\_hari = (check\_out - check\_in).days

# Menghitung total biaya

total\_biaya = selisih\_hari \* harga\_kamar \* 1 # Contoh: Total harga = harga \* 1

print(f"Total biaya adalah: {total\_biaya}")

# Tambahkan data booking ke dalam tabel booking

insert\_booking\_query = "INSERT INTO booking (id\_rooms, id\_user, check\_in, check\_out, nama, telp, cara\_bayar, total\_pembelian) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"

cursor.execute(insert\_booking\_query,

(id\_rooms, id\_user, check\_in, check\_out, nama, telp, cara\_bayar, total\_biaya))

print("Booking successful!")

# Update status kamar menjadi 'booked'

update\_status\_query = "UPDATE rooms SET status = 'booked' WHERE id\_rooms = %s"

cursor.execute(update\_status\_query, (id\_rooms,))

connection.commit()

else:

print("Kamar tidak tersedia atau ID Kamar tidak valid.")

### **Code untuk connect database**

def connect\_to\_database():

global connection # Tambahkan variabel connection global

db\_config = {

'host': 'localhost',

'user': 'root',

'password': '',

'database': 'hotel'

}

try:

connection = mysql.connector.connect(\*\*db\_config)

print("Connected to MySQL database")

return connection

except Error as err:

print(f"Error: {err}")

return None

### **Code untuk close connection**

def close\_connection(connection, cursor=None):

if cursor:

cursor.close()

if connection.is\_connected():

connection.close()

print("Connection closed")

### **Code untuk register user**

def register\_user(username, password, cursor, connection):

insert\_query = "INSERT INTO users (username, password) VALUES (%s, %s)"

cursor.execute(insert\_query, (username, password))

connection.commit()

print("Registration successful!")

### **Code untuk login**

def login\_user(username, password, cursor):

query\_user = "SELECT \* FROM users WHERE username = %s AND password = %s"

query\_admin = "SELECT \* FROM logad WHERE username = %s AND password = %s"

cursor.execute(query\_user, (username, password))

user = cursor.fetchone()

if user:

print("Login successful!")

return 'user', user[0] # Mengembalikan role dan id\_user

else:

cursor.execute(query\_admin, (username, password))

admin = cursor.fetchone()

if admin:

print("Selamat datang, Admin")

return 'admin', None

else:

print("Login failed. Invalid username or password.")

return None

### **Code untuk select room**

def select\_rooms(cursor):

select\_query = "SELECT \* FROM rooms"

cursor.execute(select\_query)

result = cursor.fetchall()

if result:

print("-" \* 100)

print("{:<15} | {:<15} | {:<15} | {:<15} | {:<15}".format(

"ID Booking", "NO Kamar", "Type Kamar", "Harga", "Lantai", "Status Kamar"

))

print("-" \* 100)

for row in result:

row = tuple('-' if item is None else item for item in row)

print("{:<15} {:<15} {:<15} {:<15} {:<15}".format(

row[0], row[1], row[2], row[3], row[4]

))

else:

print("Tidak ada data untuk ditampilkan.")

### **Code untuk menghapus kamar role admin**

def delete\_room(cursor):

id\_rooms = input("Masukkan ID Kamar yang akan dihapus: ")

delete\_query = "DELETE FROM rooms WHERE id\_rooms = %s"

cursor.execute(delete\_query, (id\_rooms,))

connection.commit()

print("Data kamar berhasil dihapus.")

### **Code untuk update room role admin**

def update\_room(cursor):

id\_rooms = input("Masukkan ID Kamar yang akan diupdate: ")

new\_price = input("Masukkan harga baru untuk kamar: ")

update\_query = "UPDATE rooms SET harga = %s WHERE id\_rooms = %s"

cursor.execute(update\_query, (new\_price, id\_rooms))

connection.commit()

print("Data kamar berhasil diupdate.")

### **Code untuk tambah data kamar role admin**

|  |
| --- |
| Code untuk tambah data kamar role admin  def tambah\_data\_kamar(cursor):  tipe\_kamar = input("Masukkan Tipe Kamar: ")  harga\_kamar = input("Masukkan Harga Kamar: ")  status\_kamar = input("Masukkan Status Kamar (Avaible/booked): ")  insert\_query = "INSERT INTO rooms (type, harga, status) VALUES (%s, %s, %s)"  cursor.execute(insert\_query, (tipe\_kamar, harga\_kamar, status\_kamar))  connection.commit()  print("Data kamar berhasil ditambahkan.") |

### **Code untuk function main**

def main():

global connection

connection = connect\_to\_database()

if connection:

try:

cursor = connection.cursor()

create\_users\_table(cursor)

while True:

print('''

"==========================================="

"| WELCOME TO HOTELA |"

"==========================================="

''')

print("1. Login")

print("2. Register")

print("3. Exit")

choice = input("Masukkan pilihan Anda (1/2/3): ")

if choice == '1':

print('''

"==========================================="

"| WELCOME TO HOTELA LOGIN |"

"==========================================="

''')

username = input("Masukkan username: ")

password = input("Masukkan password: ")

role, id\_user = login\_user(username, password, cursor)

if role:

while True:

print('''

"==========================================="

"| WELCOME TO HOTELA MENU |"

"==========================================="

''')

if role == 'admin':

print("1. Lihat Pilihan Kamar")

print("2. Booking Kamar")

print("3. Riwayat Transaksi")

print("4. Lihat Grafik Penjualan")

print("5. Tambah Data Kamar")

print("6. Update Data Kamar")

print("7. Hapus Data Kamar")

print("8. Logout")

else:

print("1. Lihat Pilihan Kamar")

print("2. Booking Kamar")

print("3. Riwayat Transaksi")

print("4. Logout")

submenu\_choice = input("Masukkan pilihan Anda: ")

print()

if role == 'admin':

if submenu\_choice == '1':

select\_rooms(cursor)

print()

elif submenu\_choice == '2':

id\_rooms = input("Masukkan ID Kamar yang akan dibooking: ")

booking\_kamar(id\_rooms, id\_user, cursor)

print()

elif submenu\_choice == '3':

tampilkan\_riwayat\_pembelian()

print()

elif submenu\_choice == '4':

tampilkan\_grafik\_penjualan()

elif submenu\_choice == '5':

tambah\_data\_kamar(cursor)

print()

elif submenu\_choice == '6':

update\_room(cursor)

print()

elif submenu\_choice == '7':

delete\_room(cursor)

print()

elif submenu\_choice == '8':

break

else:

print("Pilihan tidak valid. Silakan pilih antara 1-8.")

else:

if submenu\_choice == '1':

tampilkan\_pilihan\_kamar(role)

print()

elif submenu\_choice == '2':

id\_rooms = input("Masukkan NO Kamar yang akan dibooking: ")

booking\_kamar(id\_rooms, id\_user, cursor)

print()

elif submenu\_choice == '3':

riwayat\_pembelian\_user(id\_user, cursor)

print()

elif submenu\_choice == '4':

break

else:

print("Pilihan tidak valid. Silakan pilih antara 1-4.")

# break

elif choice == '2':

print('''

"==========================================="

"| WELCOME TO HOTELA LOGIN |"

"==========================================="

''')

print("1. Register User")

print("2. Exit Registration")

register\_choice = input("Masukkan pilihan Anda (1/2): ")

if register\_choice == '1':

registration(cursor, connection)

elif register\_choice == '2':

pass

else:

print("Pilihan tidak valid. Silakan pilih antara 1 atau 2.")

elif choice == '3':

break

finally:

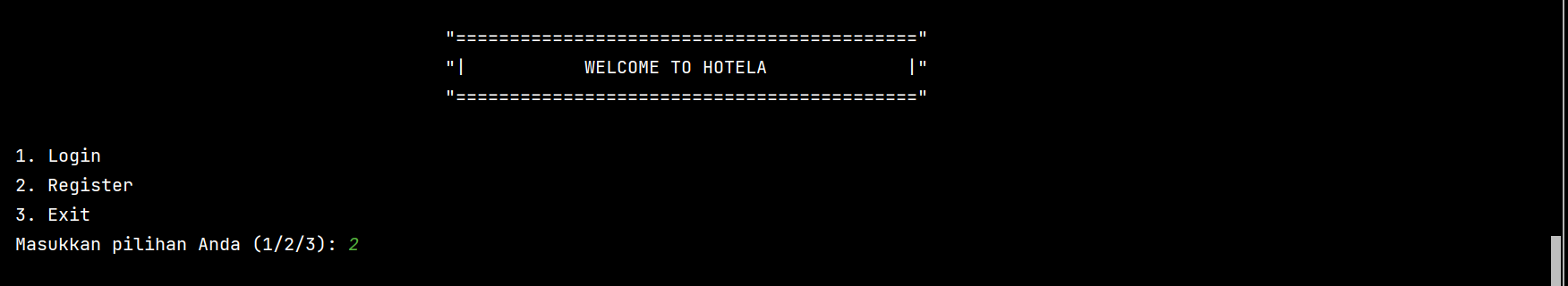
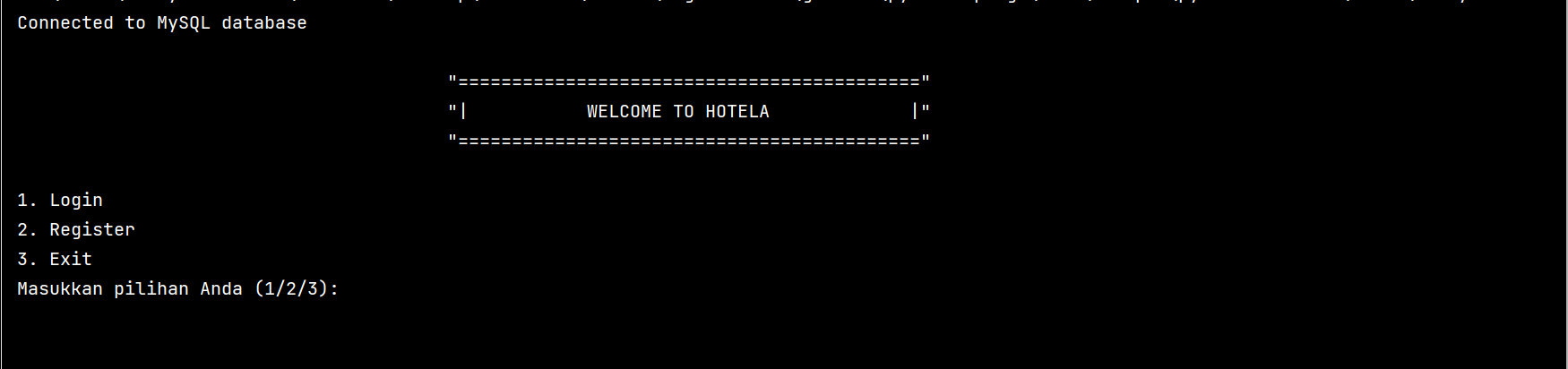
close\_connection(connection, cursor)

### **Code untuk program utama**

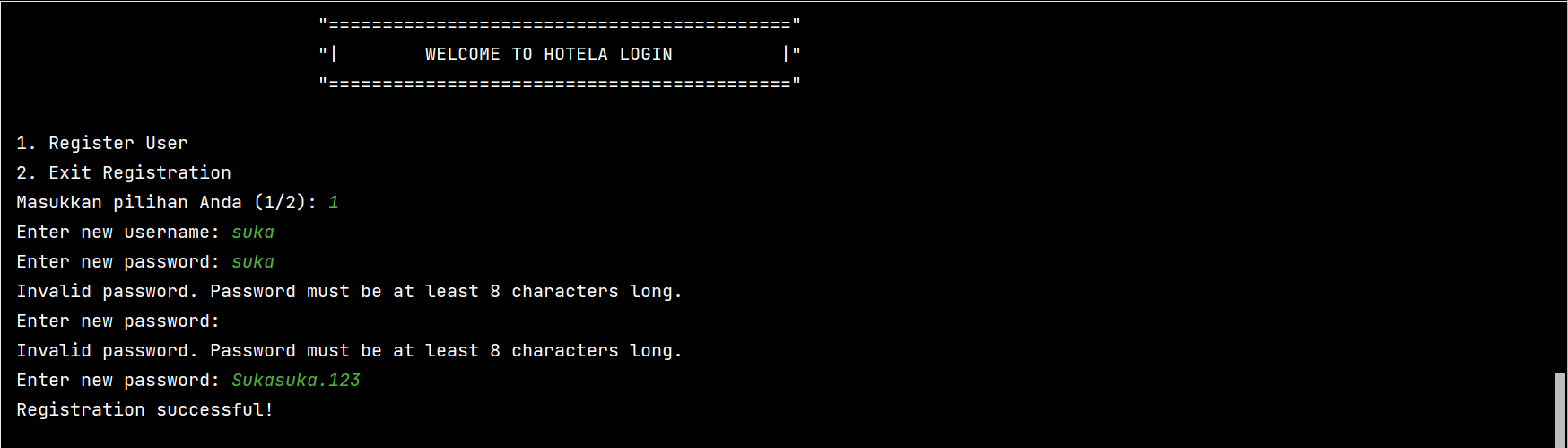
if \_\_name\_\_ == "\_\_main\_\_":

main()

## **Screenshoot Hasil Program**

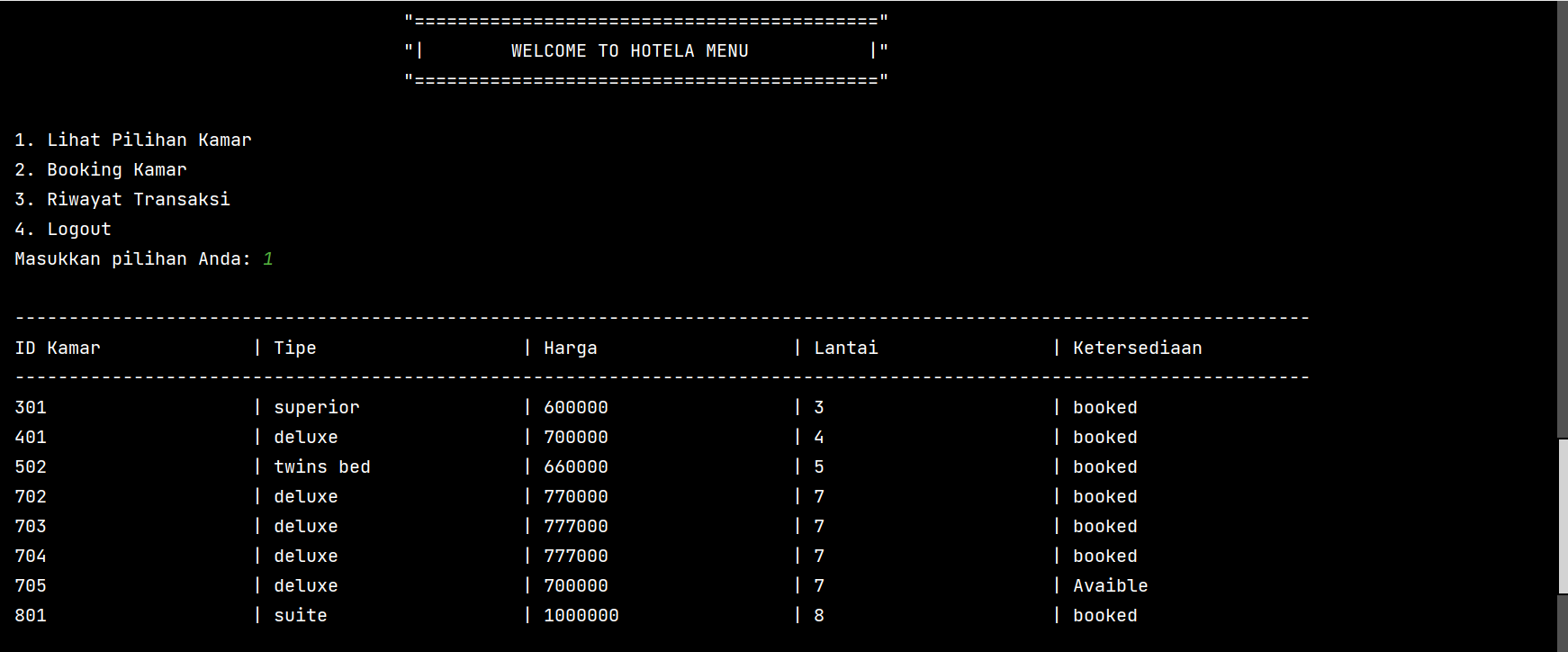
****Gambar 4.2.1 menu login

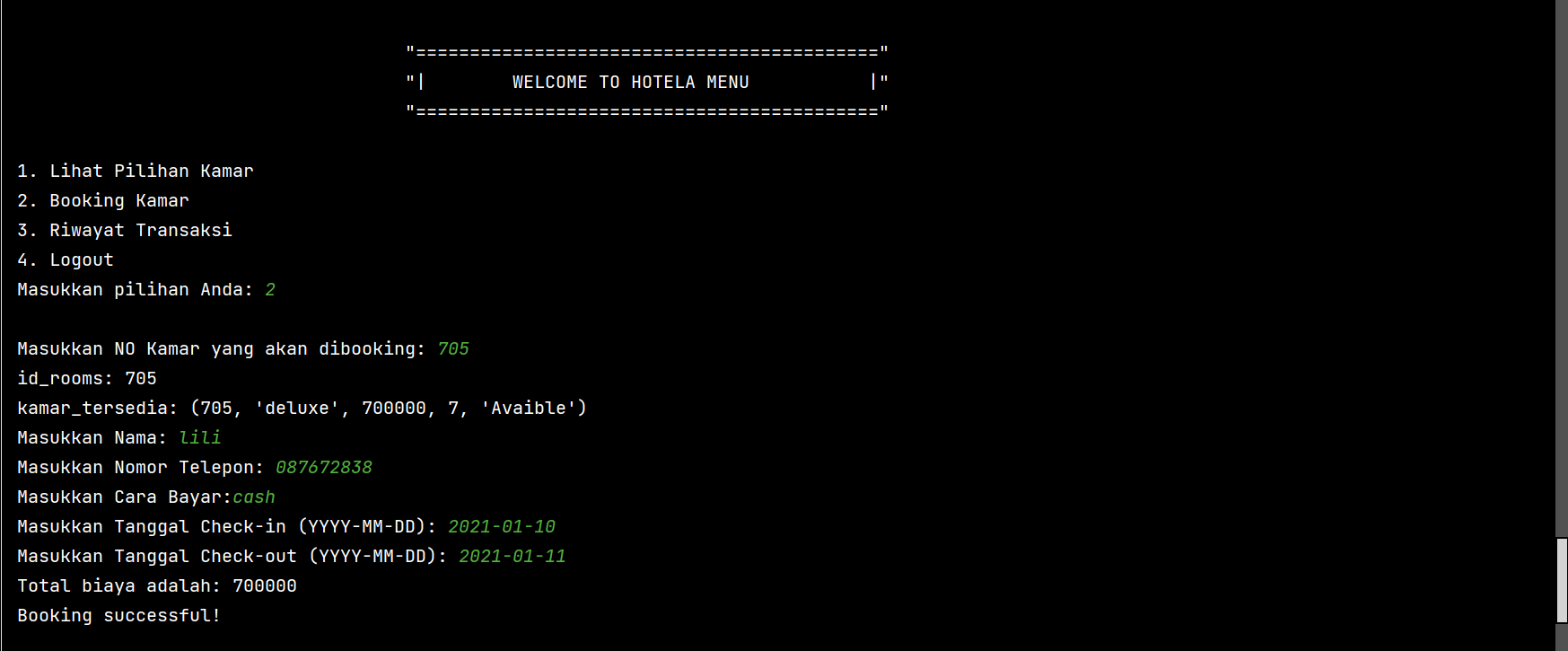
Gambar 4.2.2 jika register

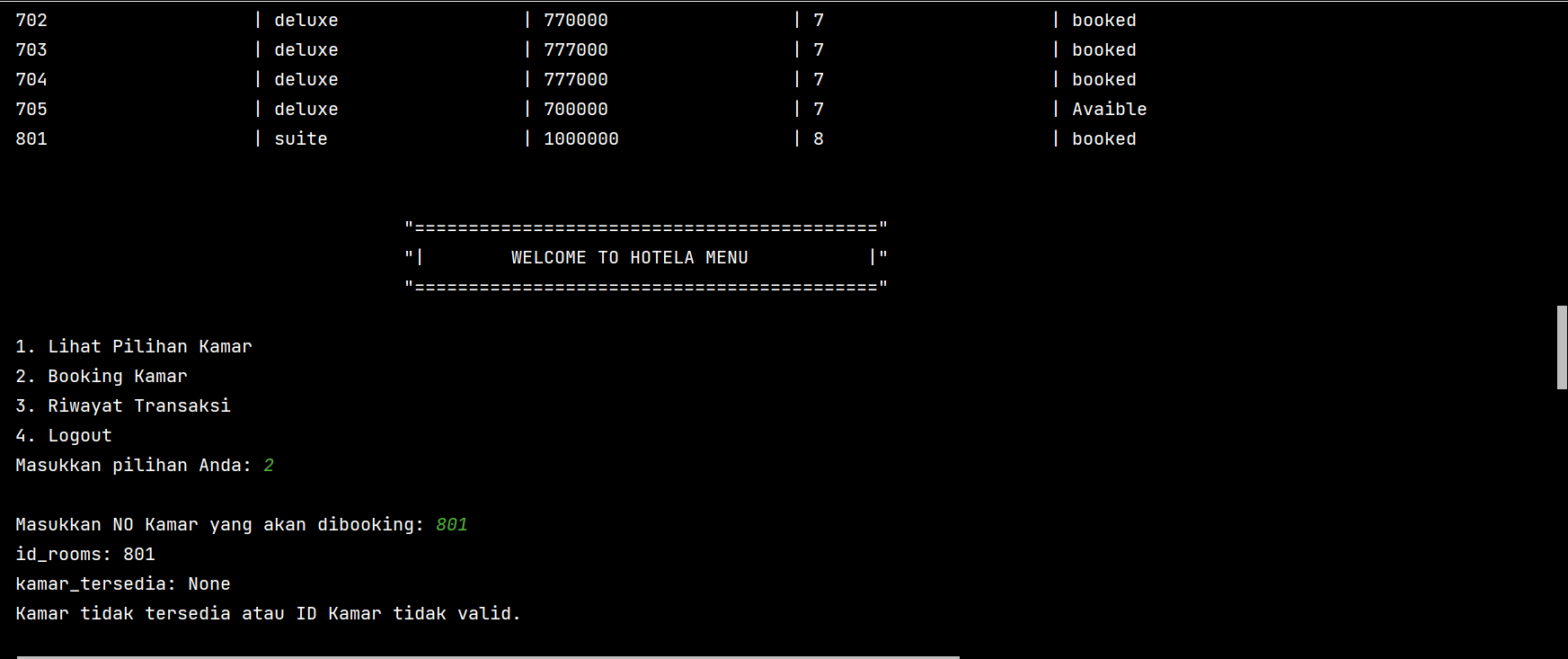


Gambar 4.2.3 menu register, jika username dan password salah atau sesuai ketentuan

Gambar 4.2.4 user memilih login

Gambar 4.2.5 tampilan menu user

Gambar 4.2.6 saat booking berhasil

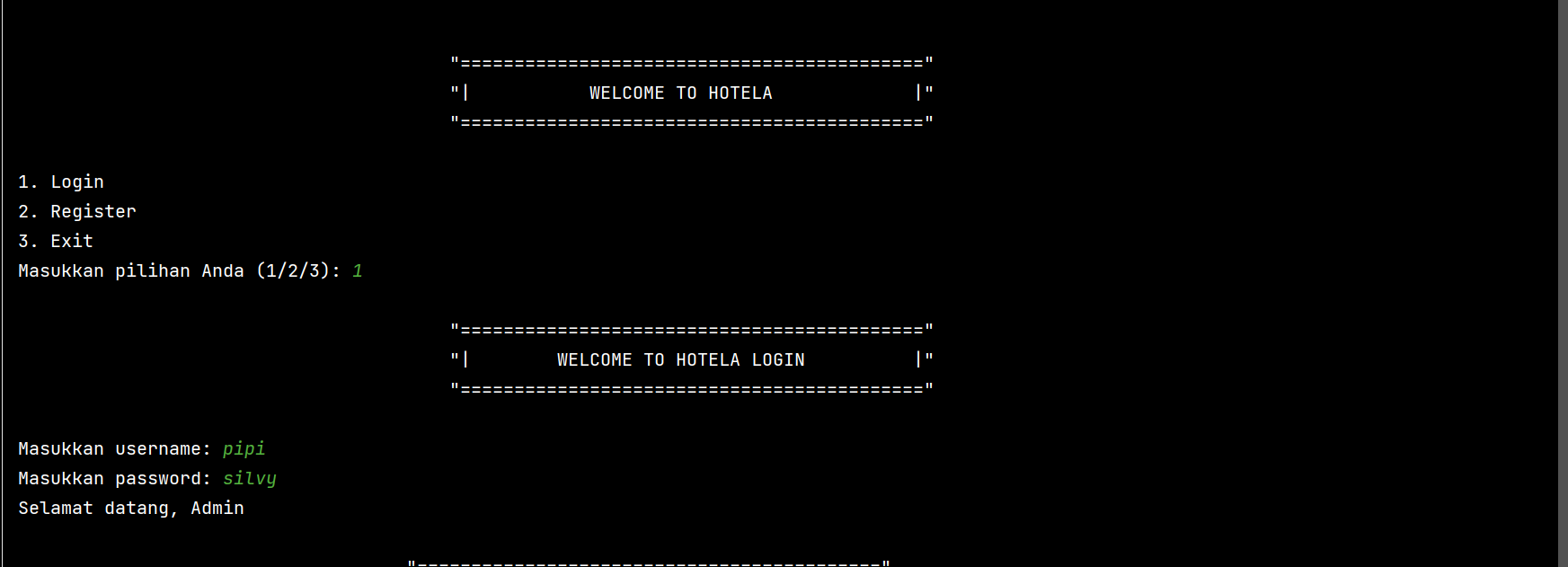


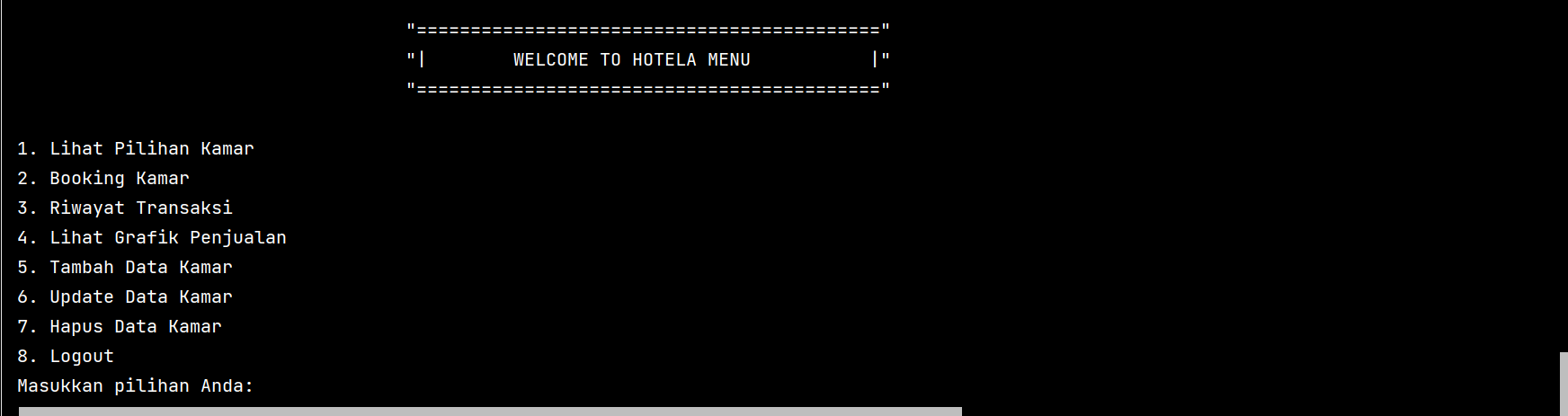
Gambar 4.2.7 saat room yang dipilih tidak tersedia

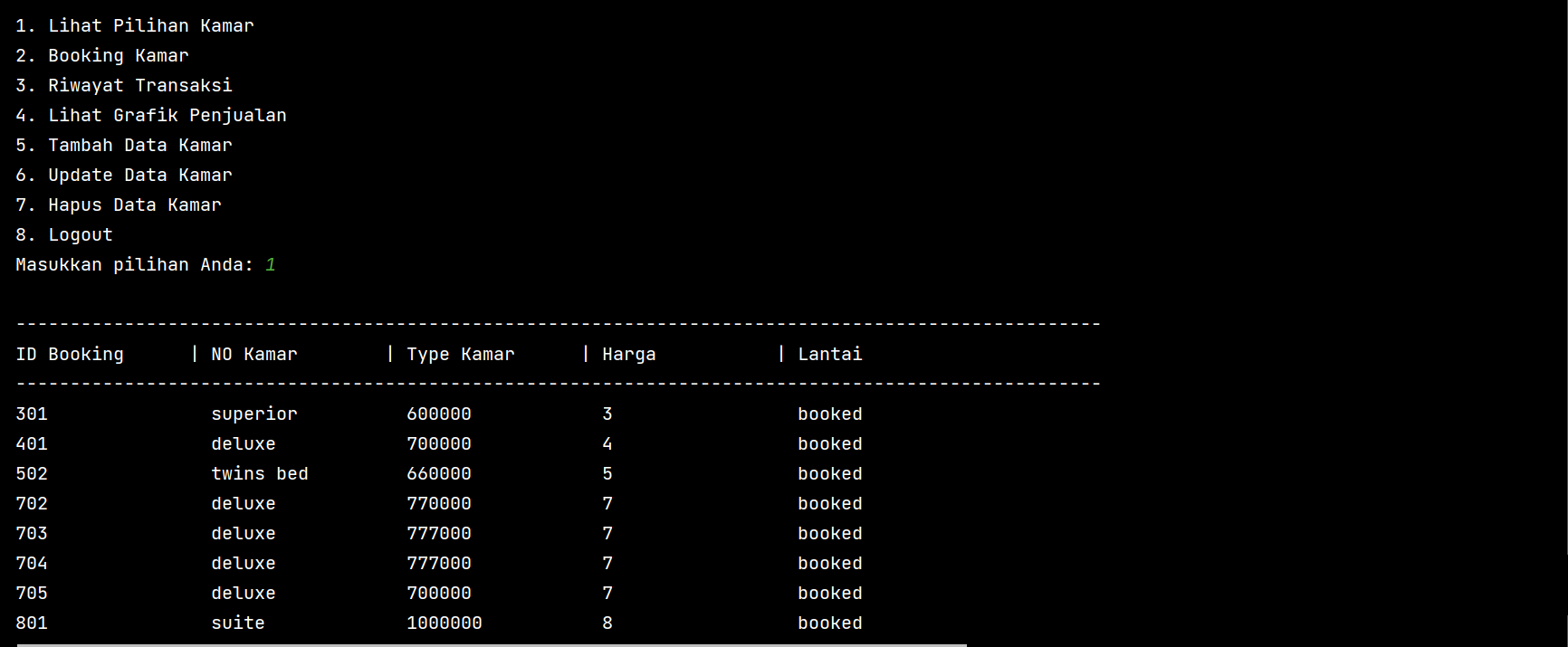


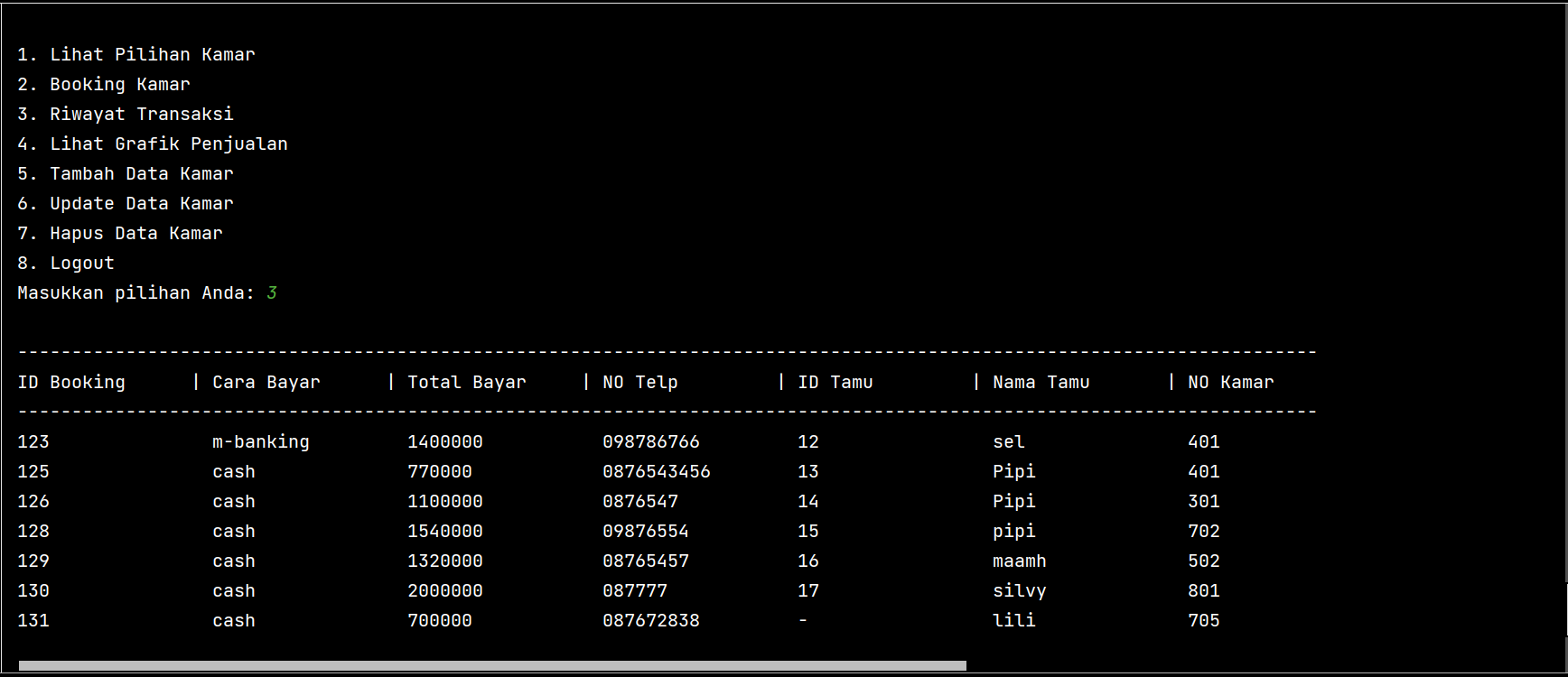
Gambar 4.2.8 riwayat transaksi user

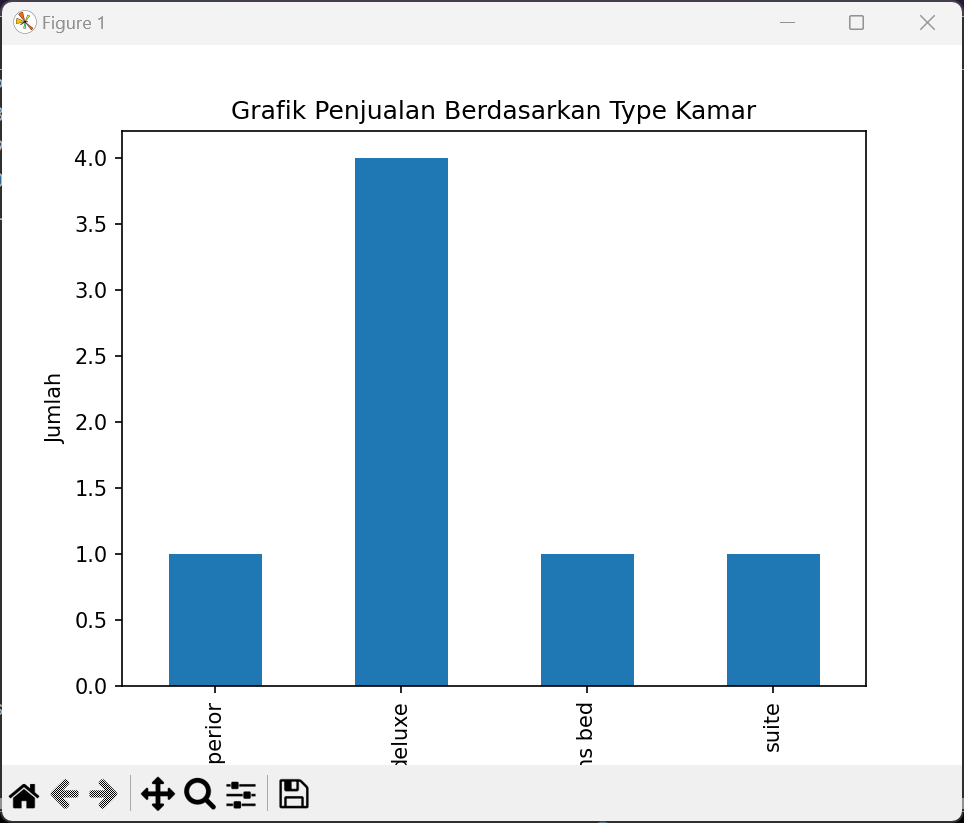
Gambar 4.2.9 saat user logout

gambar 4.2.10 login admin

Gambar 4.2.11 menu adminn

gambar 4.2.12 tampilan riwayat transaksi saat user sudah transaksi

gambar 4.2.13 riwayat transaksi admin

gambar 4.2.14 grafik penjualan admin

# BAB V

# PENUTUP

# KESIMPULAN

Berdasarkan implementasi program yang telah dibuat, dapat disimpulkan bahwa Hotela merupakan aplikasi sederhana untuk manajemen pemesanan kamar hotel. Aplikasi ini memiliki beberapa fitur utama, termasuk login, pilihan kamar, booking kamar, riwayat transaksi, grafik penjualan, dan manajemen data kamar.

Pengguna aplikasi dapat login sebagai pengguna (user) atau administrator (admin). Pengguna dapat melihat pilihan kamar, melakukan booking kamar, dan melihat riwayat transaksi pemesanan kamar mereka. Di sisi lain, administrator dapat melihat pilihan kamar, melakukan booking kamar atas permintaan pengguna, melihat riwayat transaksi secara keseluruhan, melihat grafik penjualan berdasarkan tipe kamar, serta mengelola data kamar seperti menambah, mengubah, atau menghapus informasi kamar.

Dalam pengembangan program ini, digunakan database MySQL untuk menyimpan data pengguna, data kamar, dan riwayat transaksi. Aplikasi juga memanfaatkan konsep fungsi dan modularitas untuk memudahkan pengembangan dan pemeliharaan kode.

Meskipun program ini masih sederhana, namun memberikan dasar yang baik untuk pengembangan lebih lanjut. Beberapa perbaikan dan penambahan fitur mungkin dapat dilakukan, seperti penanganan error yang lebih baik, validasi input yang lebih ketat, dan peningkatan antarmuka pengguna.

Dengan demikian, Hotela memberikan solusi dasar untuk manajemen pemesanan kamar hotel dengan menggunakan bahasa pemrograman Python dan database MySQL. Program ini dapat menjadi landasan untuk pengembangan aplikasi manajemen hotel yang lebih kompleks dan dapat disesuaikan dengan kebutuhan pengguna atau pemilik hotel.