

**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 kehadiran 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 ucapan 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

<b>BAB I.....</b>	<b>1</b>
<b>1.1 Latar Belakang .....</b>	<b>1</b>
<b>1.2 Rumusan Masalah.....</b>	<b>1</b>
<b>1.3 Tujuan .....</b>	<b>1</b>
<b>1.4 Manfaat .....</b>	<b>2</b>
<b>1.5 Sistematika Penulisan .....</b>	<b>2</b>
<b>BAB II.....</b>	<b>3</b>
<b>LANDASAN TEORI.....</b>	<b>3</b>
<b>2.1 Software yang digunakan .....</b>	<b>3</b>
<b>2.2 Bahasa Pemrograman yang Digunakan.....</b>	<b>4</b>
<b>2.3 Kondisional (if-else).....</b>	<b>5</b>
<b>2.4 Array .....</b>	<b>5</b>
<b>2.5 Looping for.....</b>	<b>5</b>
<b>2.6 Looping While True .....</b>	<b>5</b>
<b>2.7 Library yang digunakan.....</b>	<b>5</b>
<b>BAB III.....</b>	<b>6</b>
<b>3.1 Fitur pada aplikasi .....</b>	<b>6</b>
<b>4.2.1 Halaman login .....</b>	<b>6</b>
<b>4.2.2 Halaman Home Admin.....</b>	<b>6</b>
<b>4.2.3 Halaman Home User.....</b>	<b>6</b>
<b>4.2.4 Pilihan Kamar .....</b>	<b>6</b>
<b>4.2.5 Booking Kamar .....</b>	<b>6</b>
<b>4.2.6 Riwayat Transaksi. ....</b>	<b>6</b>
<b>4.2.7 Grafik Penjualan.....</b>	<b>7</b>
<b>4.2.8 Manajemen Data Kamar.....</b>	<b>7</b>
<b>4.2.9 Logout .....</b>	<b>7</b>
<b>3.2 Flowchart .....</b>	<b>8</b>
<b>3.2.1 Function select.....</b>	<b>8</b>
<b>3.2.2 Function insert .....</b>	<b>9</b>
<b>3.2.3 Funtion delete .....</b>	<b>10</b>
<b>3.2.4 Funtion register .....</b>	<b>11</b>

3.2.5	Function validasi password .....	11
3.2.6	Function validasi password .....	11
3.2.7	Function login.....	12
3.2.8	Function Select Room .....	13
3.2.9	Function tambah data kamar .....	14
3.2.10	Function update data .....	15
3.2.11	Function delete kamar .....	16
3.2.12	Function untuk menambahkan data ke riwayat.....	16
3.2.13	Function riwayat pembelian .....	17
3.2.14	Function riwayat pembelian user .....	18
3.2.15	Function untuk menampilkan pilihan kamar .....	19
3.2.16	Function menampilkan grafik penjualan .....	20
3.2.17	Function untuk booking kamar .....	21
3.2.18	Lanjutan function untuk booking kamar .....	22
3.2.19	Function untuk menu login.....	23
3.2.20	Function untuk main program .....	24
3.2.21	Lanjutan function untuk main program .....	25
3.2.22	Function untuk menambahkan user tidak kali registrasi .....	26
3.2.23	Program utama .....	26
<b>BAB IV</b>	.....	<b>27</b>
<b>4.1</b>	Source code .....	<b>27</b>
4.1.1	Library dan bagian atas pada kode .....	27
4.1.2	Code untuk perintah tampilkan .....	27
4.1.3	Code untuk menambahkan data pada database.....	27
4.1.4	Code untuk mengupdate data tertentu pada database .....	27
4.1.5	Code untuk menghapus data dalam tabel database .....	27
4.1.6	Code untuk menambahkan riwayat booking.....	28
4.1.7	Code untuk menampilkan riwayat booking role admin .....	28
4.1.8	Code untuk menampilkan riwayat booking user.....	28
4.1.9	Code untuk menampilkan grafik penjualan role admin.....	28
4.1.10	Code untuk validasi password saat registrasi .....	29
4.1.11	Code untuk validasi password saat registrasi .....	29
4.1.12	Code untuk registrasi .....	29
4.1.13	Code untuk select kamar .....	29

4.1.14	Code untuk membuat row baru di tabel user jika ada yang registrasi .....	29
4.1.15	Code untuk booking kamar .....	30
4.1.16	Code untuk connect database .....	30
4.1.17	Code untuk close connection.....	31
4.1.18	Code untuk register user .....	31
4.1.19	Code untuk login .....	31
4.1.20	Code untuk select room .....	31
4.1.21	Code untuk menghapus kamar role admin .....	31
4.1.22	Code untuk update room role admin .....	32
4.1.23	Code untuk tambah data kamar role admin .....	32
4.1.24	Code untuk function main.....	32
4.1.25	Code untuk program utama .....	34
4.2	Screenshoot Hasil Program .....	34
<b>BAB V .....</b>		<b>39</b>
<b>PENUTUP.....</b>		<b>39</b>
<b>KESIMPULAN.....</b>		<b>39</b>

# **BAB I**

## **PENDAHULUAN**

### **1.1 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.

### **1.2 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?

### **1.3 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.

## **1.4 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.

## **1.5 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 menjelaskan tentang software yang digunakan dan bahasan 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 sourcecode dan screenshot 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**

#### **2.1 Software yang digunakan**

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

a. Kelebihan Pycharm :

1. menyediakan editor kode yang cerdas dengan fitur penyorotan sintaks, autocompletion, dan refactorings yang dapat meningkatkan produktivitas pengembang.
2. Tersedia untuk berbagai platform termasuk Windows, macOS, dan Linux.
3. PyCharm menyediakan dukungan terintegrasi untuk virtual environment, memungkinkan pengembang membuat dan mengelola lingkungan Python yang terisolasi.
4. Memiliki fungsi SQL editor dan dukungan database yang dapat diakses langsung dari IDE, memudahkan pengembangan aplikasi yang melibatkan database.

b. Kekurangan Pycharm :

1. Untuk proyek-proyek besar, waktu pembukaan awal proyek mungkin memakan waktu yang cukup lama.
2. PyCharm memiliki banyak fitur yang memerlukan pembelajaran awal, dan pengguna mungkin perlu waktu untuk menguasai semua fitur tersebut.
3. 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.



## 2.2 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

### a. Kelebihan bahasa Python

1. Python didesain dengan sintaks yang bersifat mudah dibaca dan dimengerti, membuatnya cocok untuk pemula maupun pengembang berpengalaman.
2. Python memiliki library yang kuat seperti Pandas, NumPy, dan Matplotlib yang mendukung analisis data, ilmu data, dan visualisasi data.
3. Sintaksis bersifat ekspresif dan pemeliharaan kode relatif mudah, memungkinkan pengembang untuk mengelola dan memperbarui proyek dengan efisien.
4. Python telah digunakan dalam pengembangan perangkat lunak berskala besar dan diterapkan dalam berbagai proyek besar dan organisasi.

### b. Kekurangan bahasa Python

1. Python dapat memakan banyak memori, terutama pada aplikasi yang membutuhkan sumber daya tinggi.
2. Untuk aplikasi yang membutuhkan kecepatan pemrosesan tinggi atau memiliki persyaratan kecepatan real-time yang ketat, Python mungkin bukan pilihan yang ideal.
3. Python adalah bahasa yang tidak bersifat statis dan tidak menerapkan tipe data yang ketat, yang dapat menyebabkan kesalahan saat runtime.

## 2.3 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).

## 2.4 Array

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

## 2.5 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

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

Gambar 2.5.1 Contoh penggunaan looping for

## 2.6 Looping While True

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

```
while True:
    print('')
```

Gambar 2.6.1 contoh penggunaan looping While True

## 2.7 Library yang digunakan

Dalam aplikasi yang dibuat 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.pyplot" : 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**

#### **3.1 Fitur pada aplikasi**

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

##### **4.2.1 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.

##### **4.2.2 Halaman Home Admin**

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

##### **4.2.3 Halaman Home User**

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

##### **4.2.4 Pilihan Kamar**

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

##### **4.2.5 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.

##### **4.2.6 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”.

#### **4.2.7 Grafik Penjualan**

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

#### **4.2.8 Manajemen Data Kamar**

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

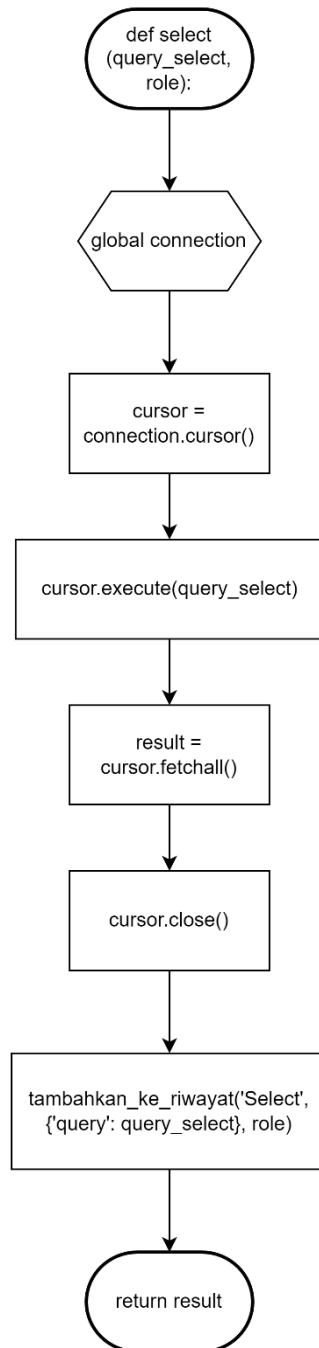
#### **4.2.9 Logout**

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

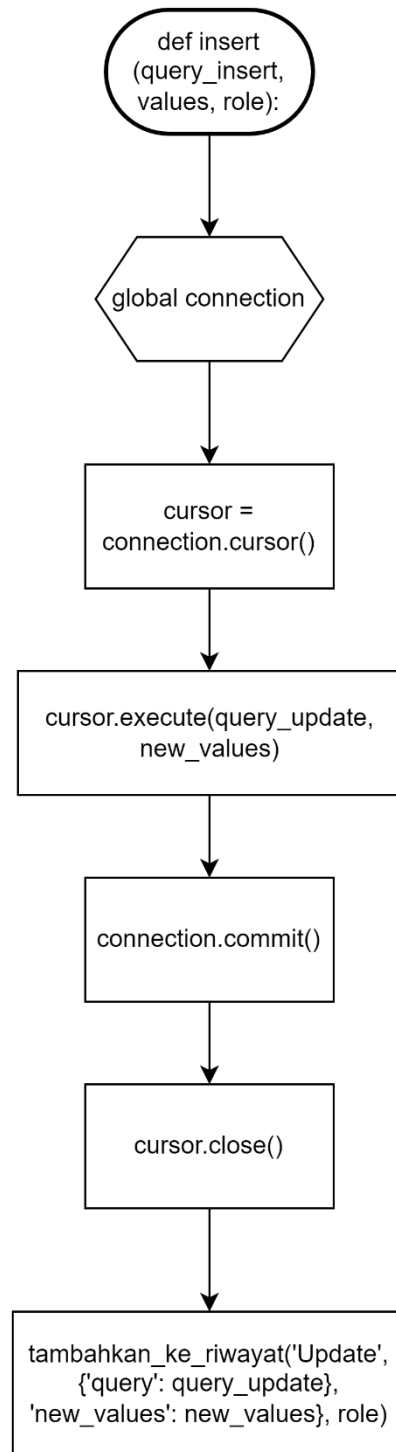
### 3.2 Flowchart

Berikut merupakan flowchart yang dirancang untuk aplikasi booking hotel, HOTELA

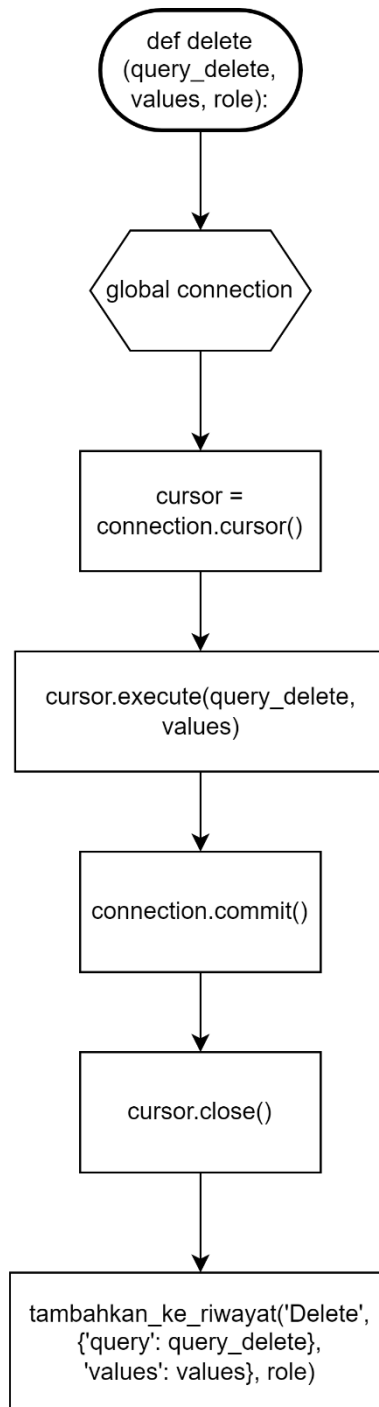
#### 3.2.1 Function select



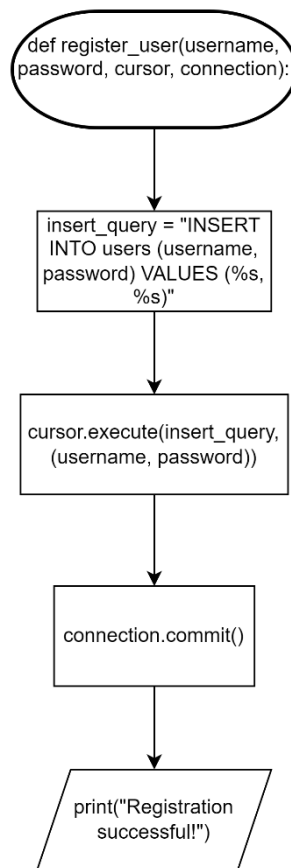
### 3.2.2 Function insert



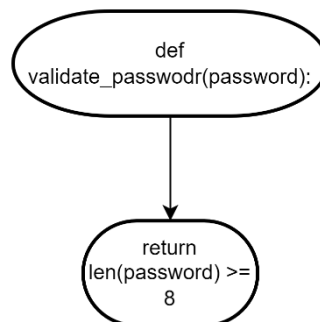
### 3.2.3 Funtion delete



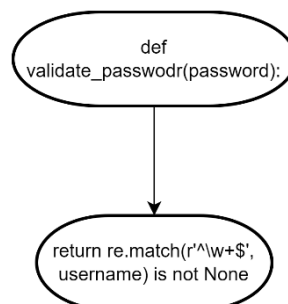
### 3.2.4 Funtion register



### 3.2.5 Function validasi password

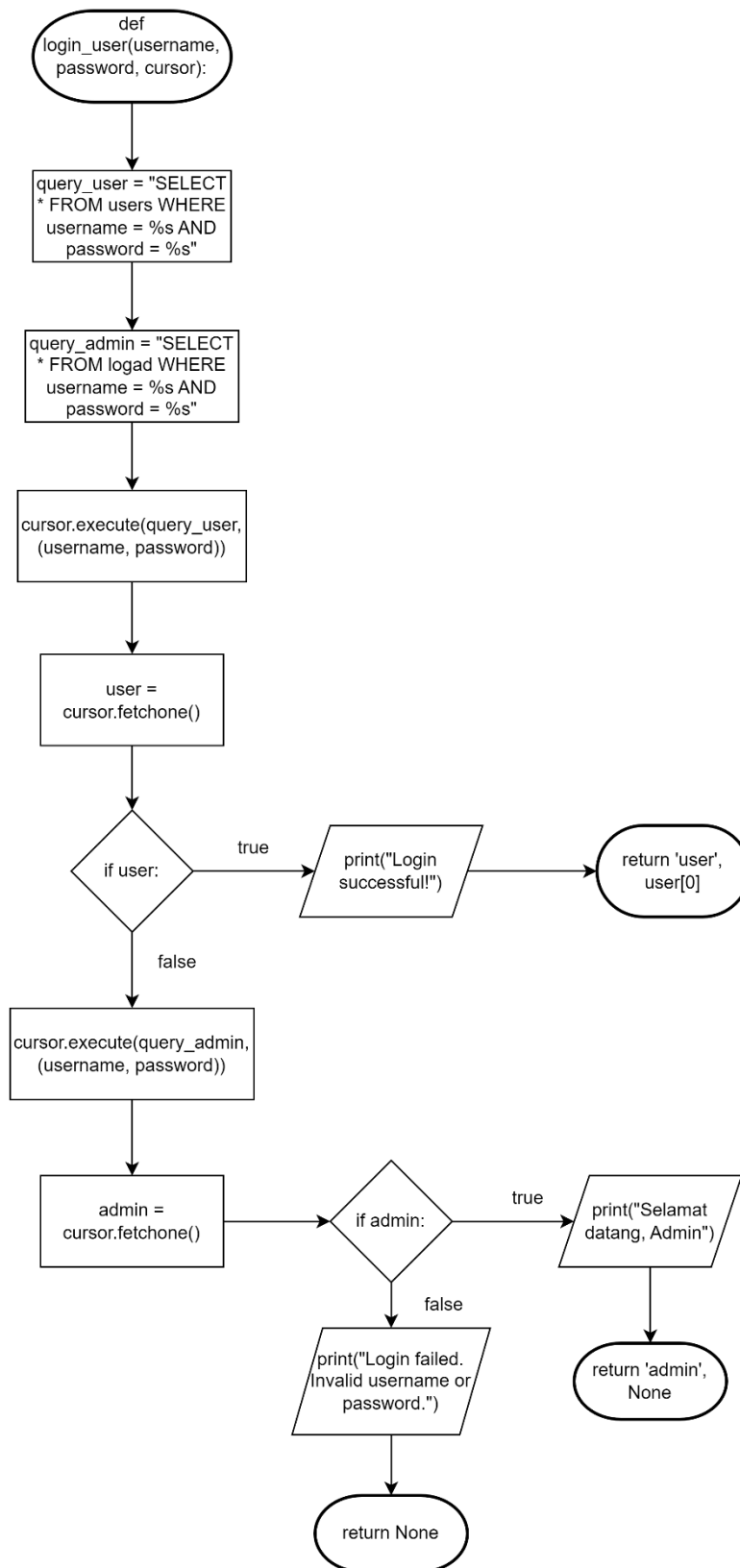


### 3.2.6 Function validasi password

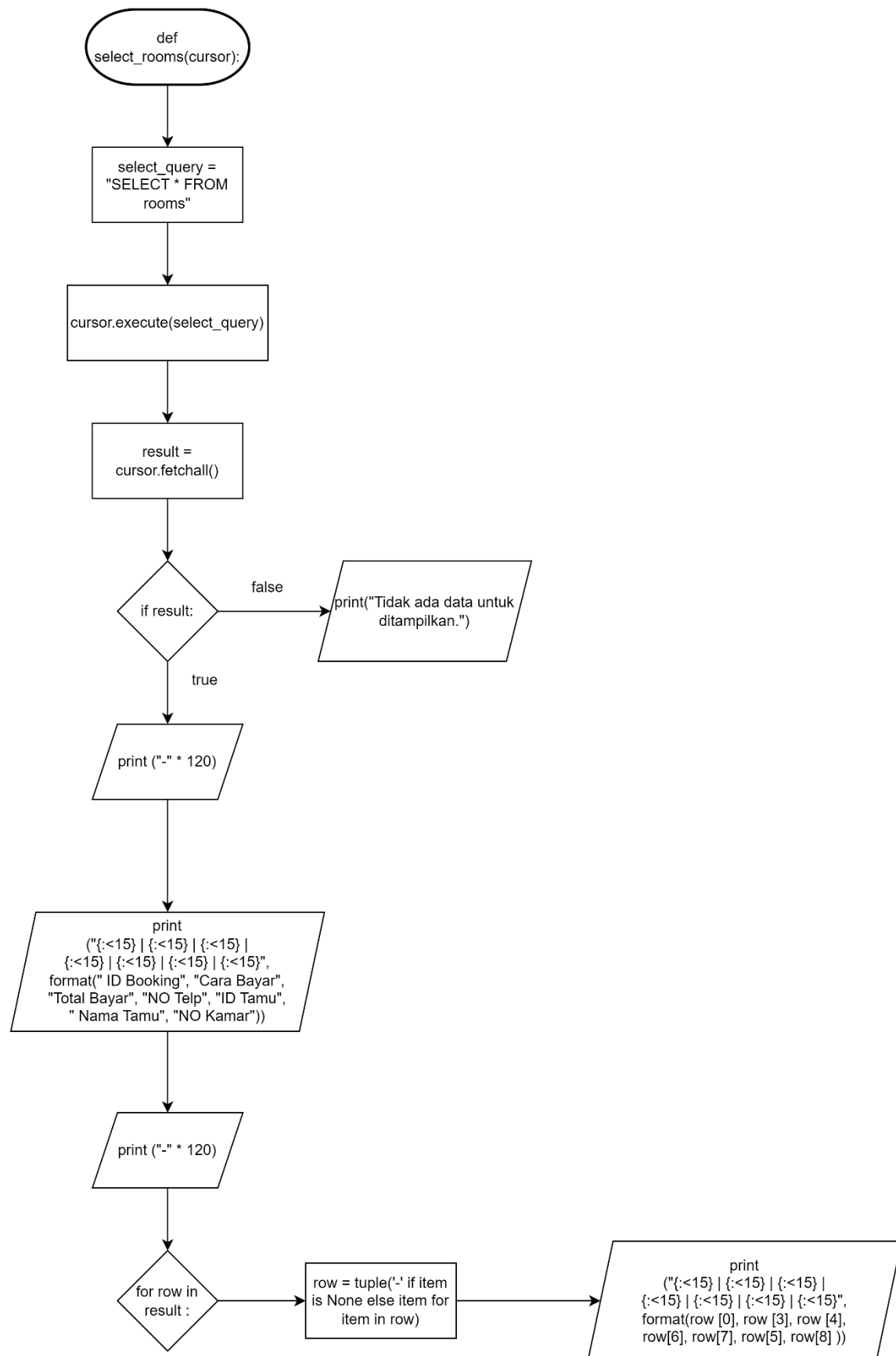




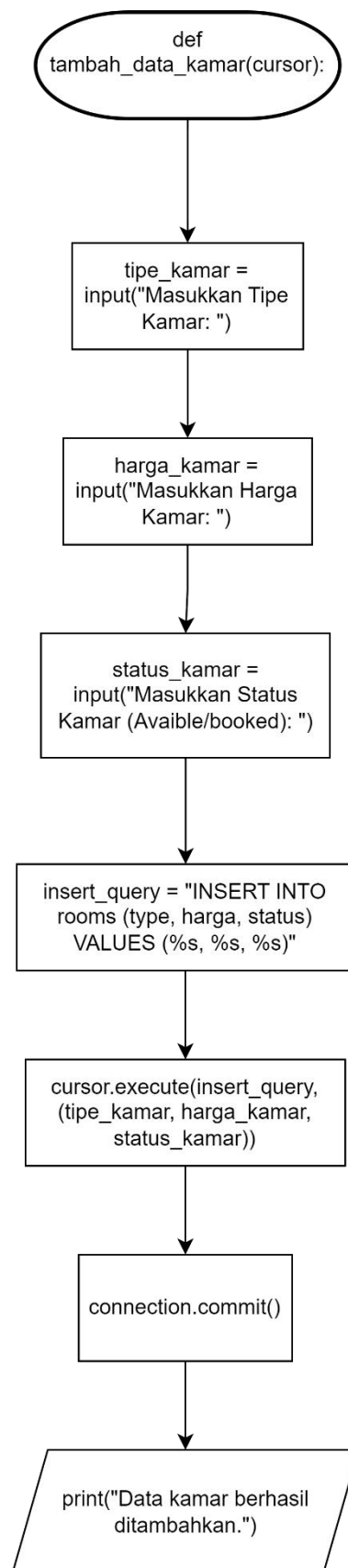
### 3.2.7 Function login



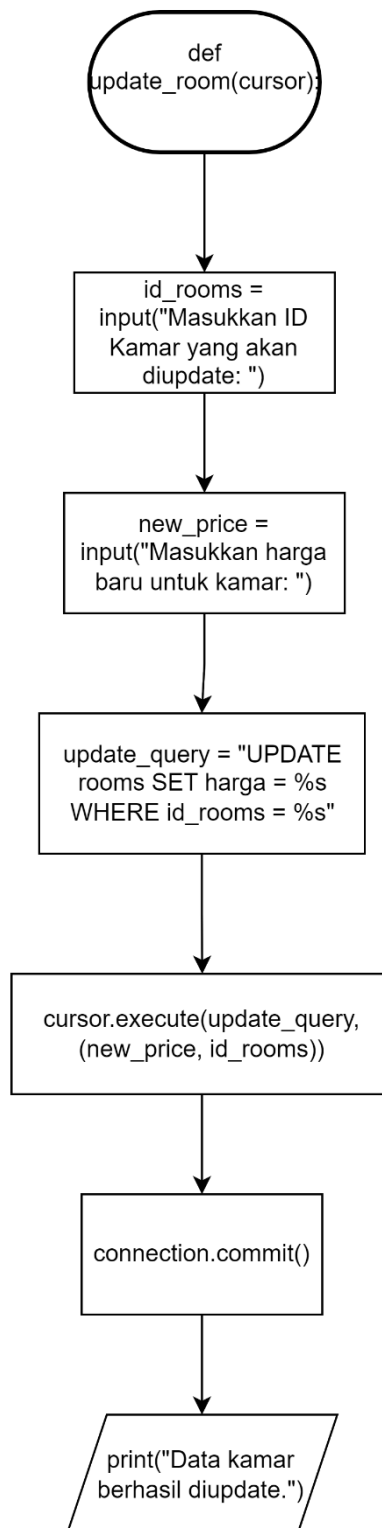
### 3.2.8 Function Select Room



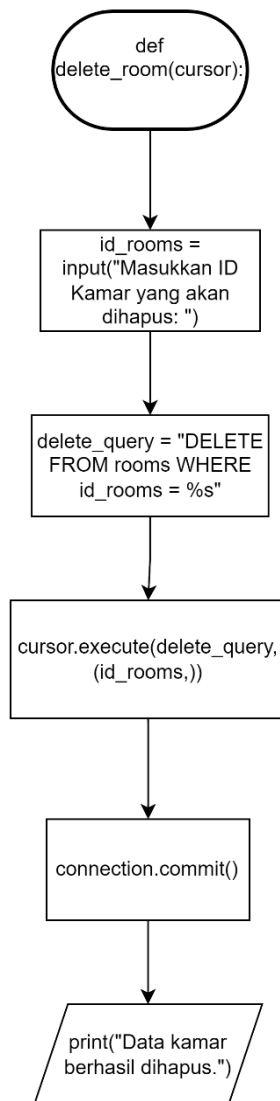
### 3.2.9 Function tambah data kamar



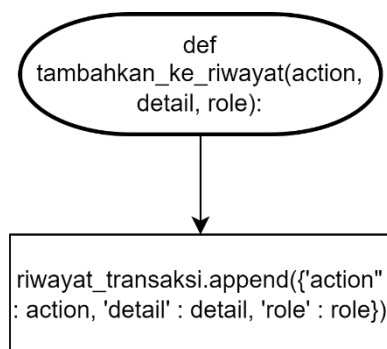
### 3.2.10 Functin update data



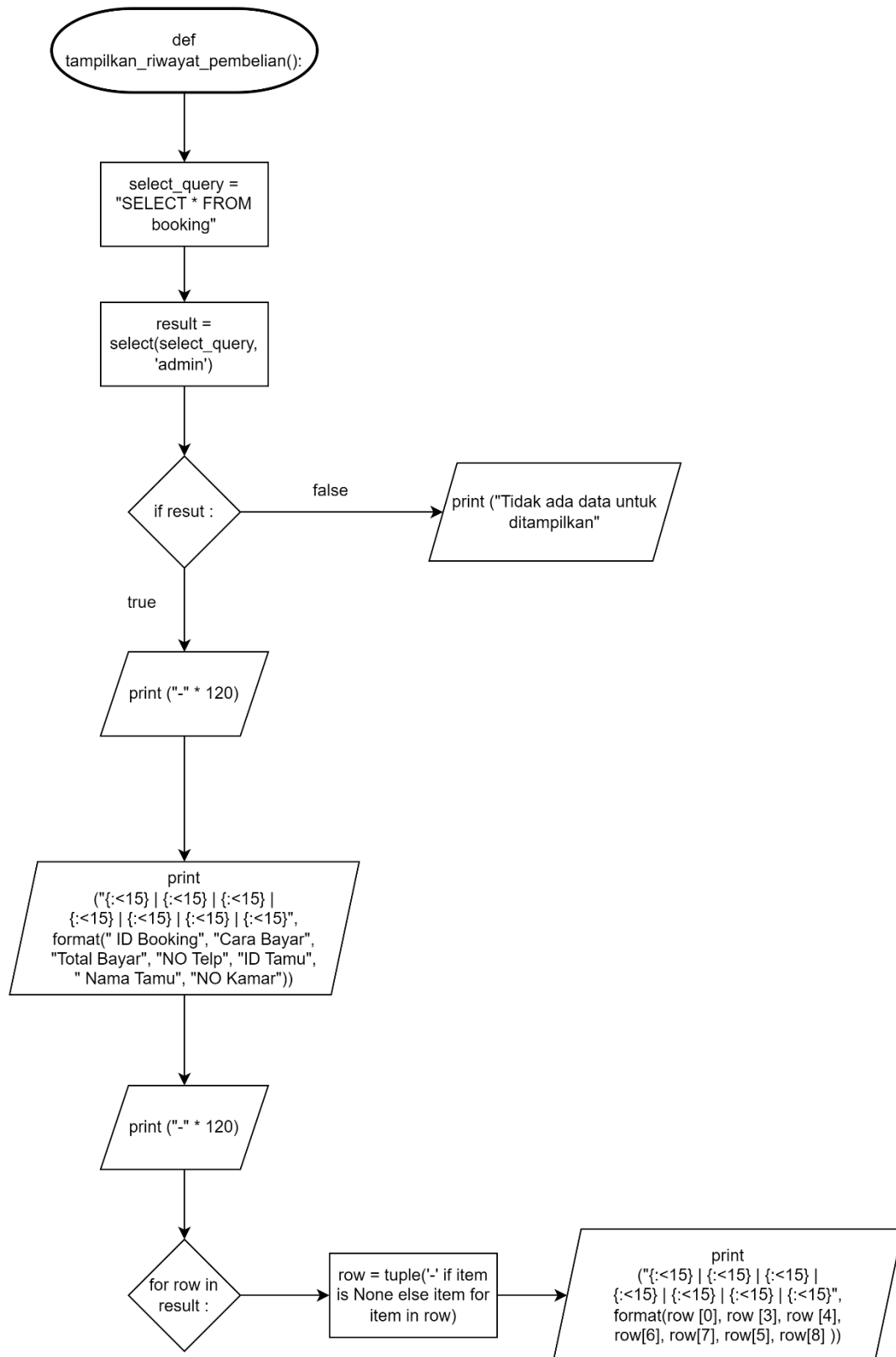
### 3.2.11 Function delete kamar



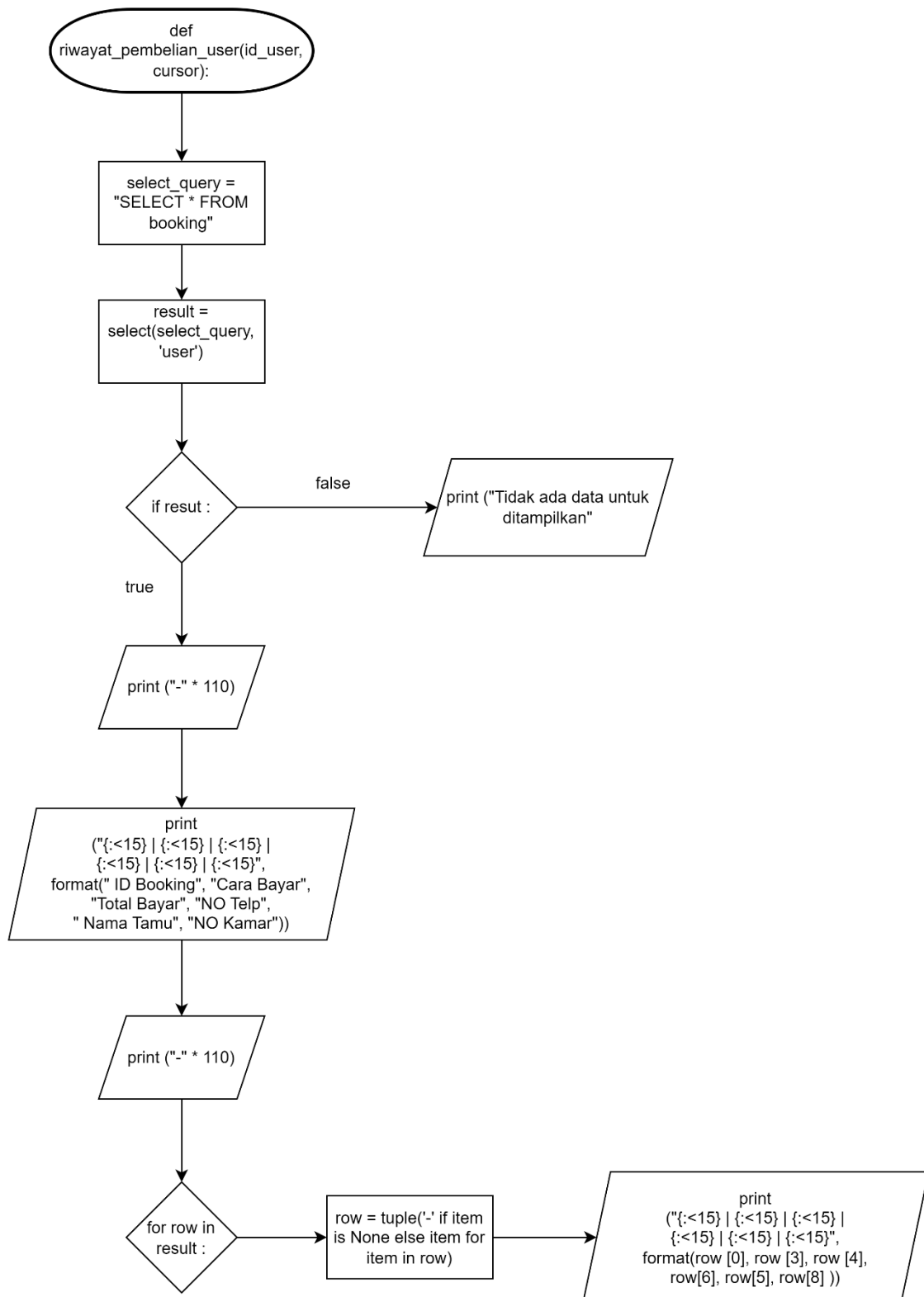
### 3.2.12 Function untuk menambahkan data ke riwayat



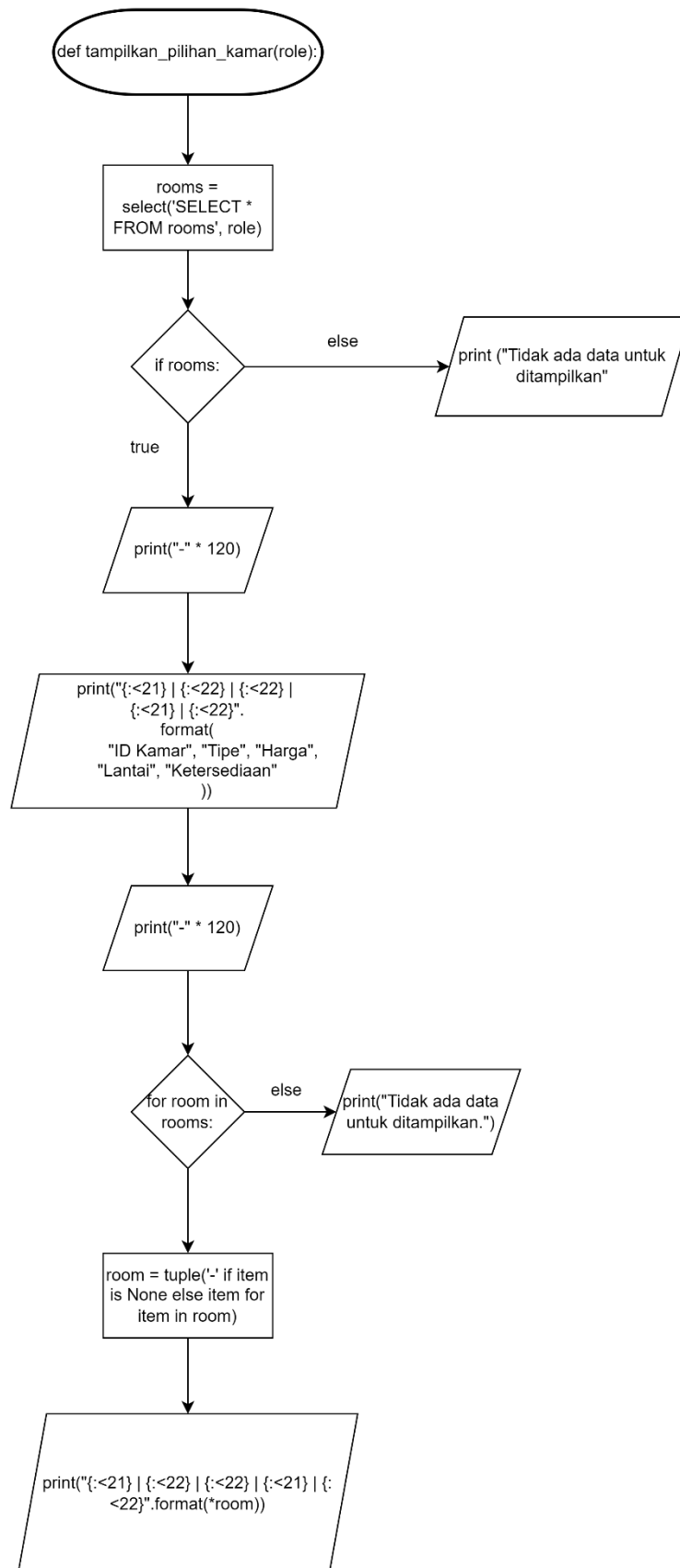
### 3.2.13 Function riwayat pembelian



### 3.2.14 Funtion riwayat pembelian user

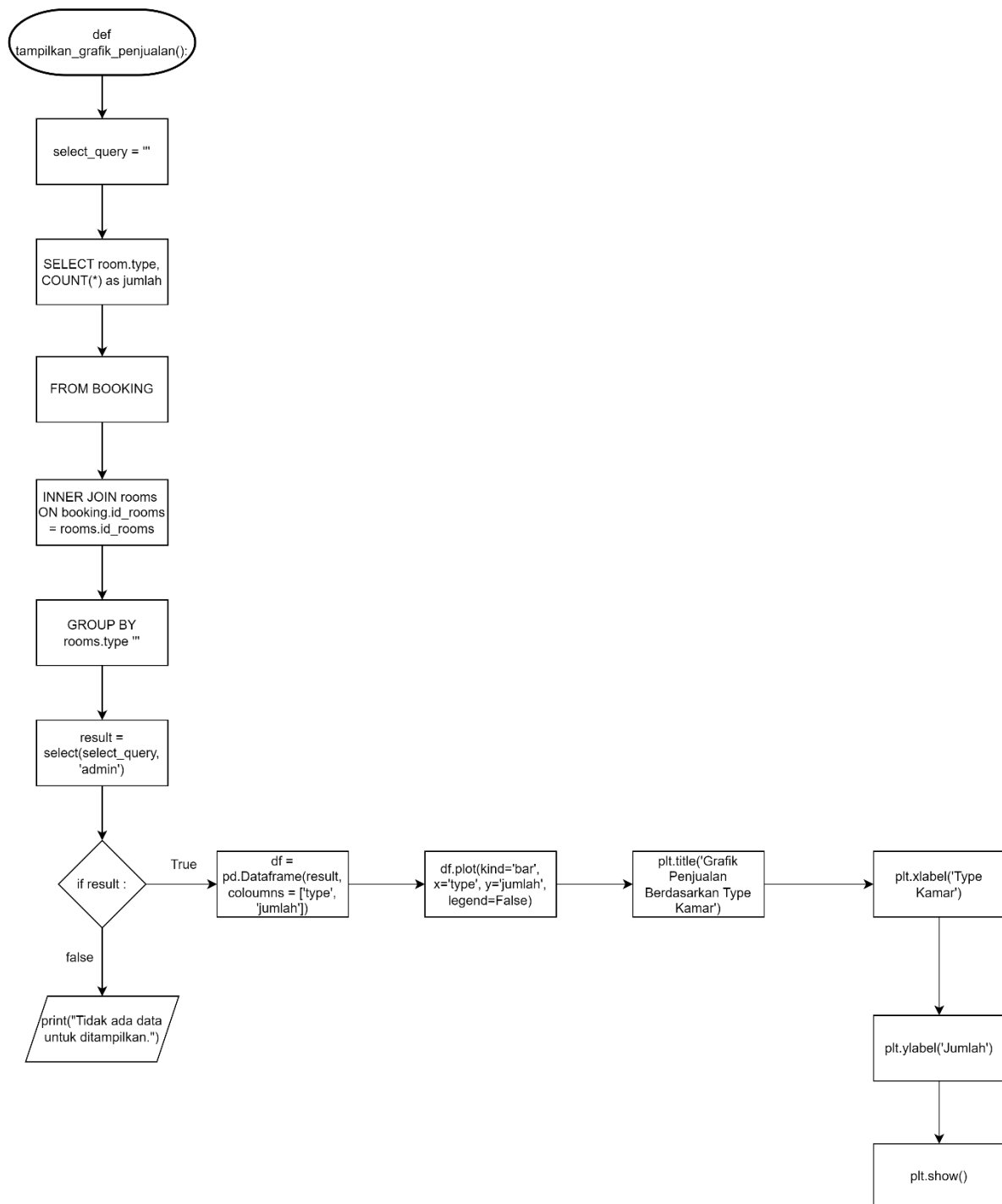


### 3.2.15 Function untuk menampilkan pilihan kamar

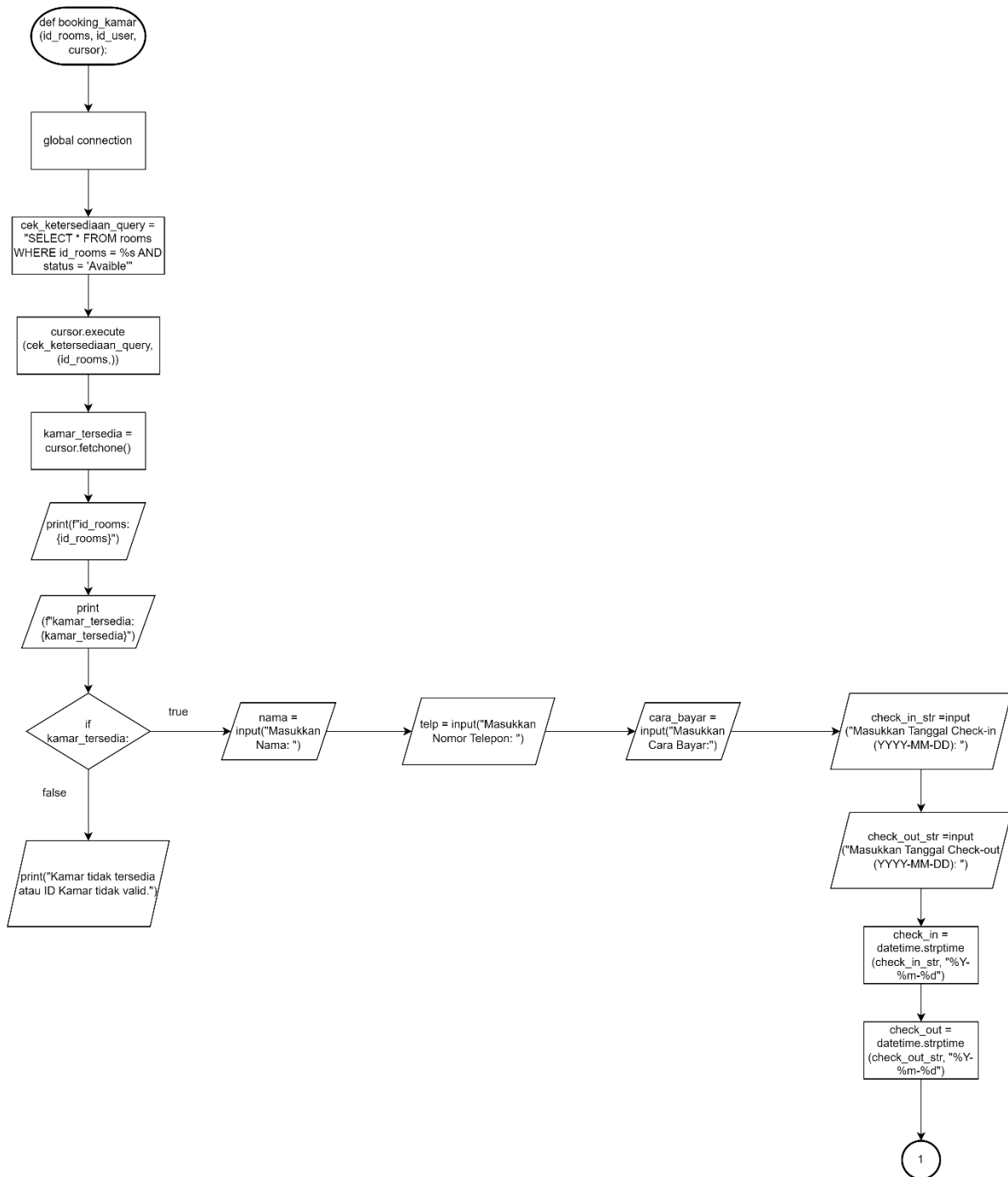




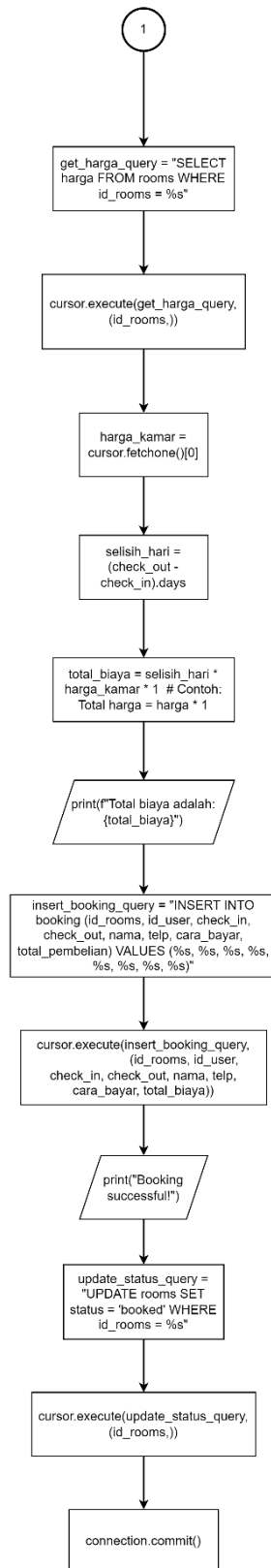
### 3.2.16 Function menampilkan grafik penjualan



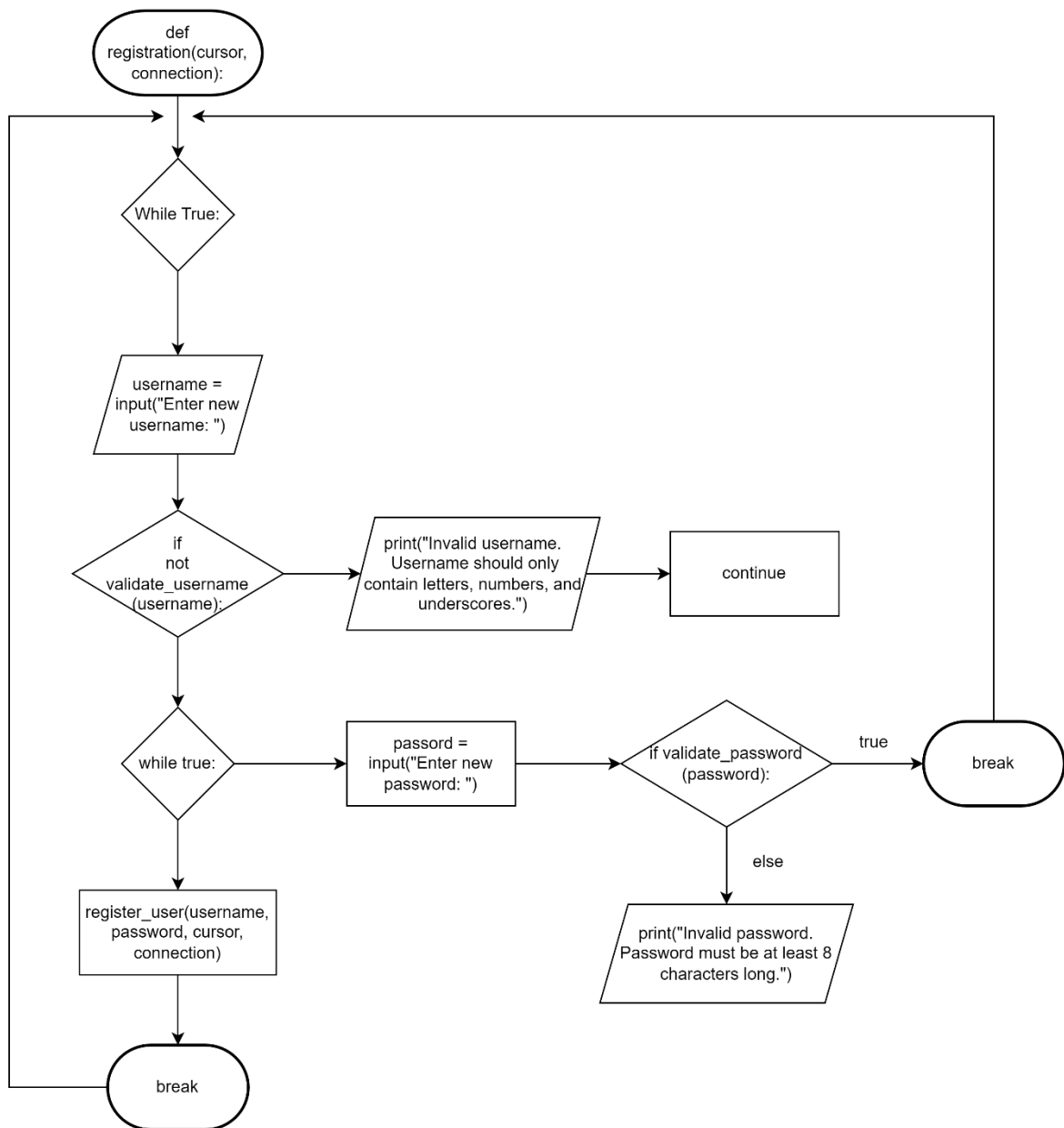
### 3.2.17 Function untuk booking kamar



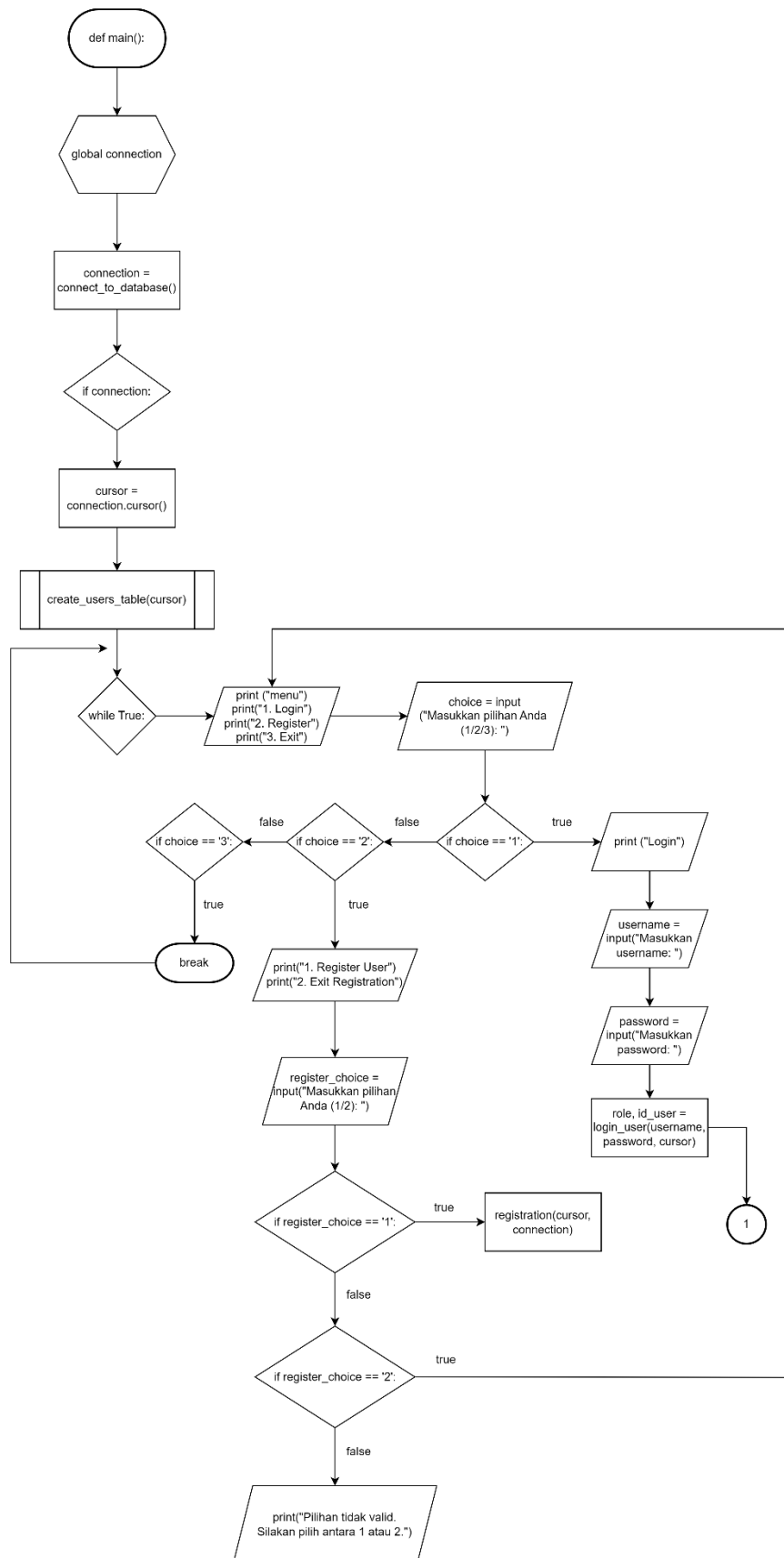
### 3.2.18 Lanjutan function untuk booking kamar



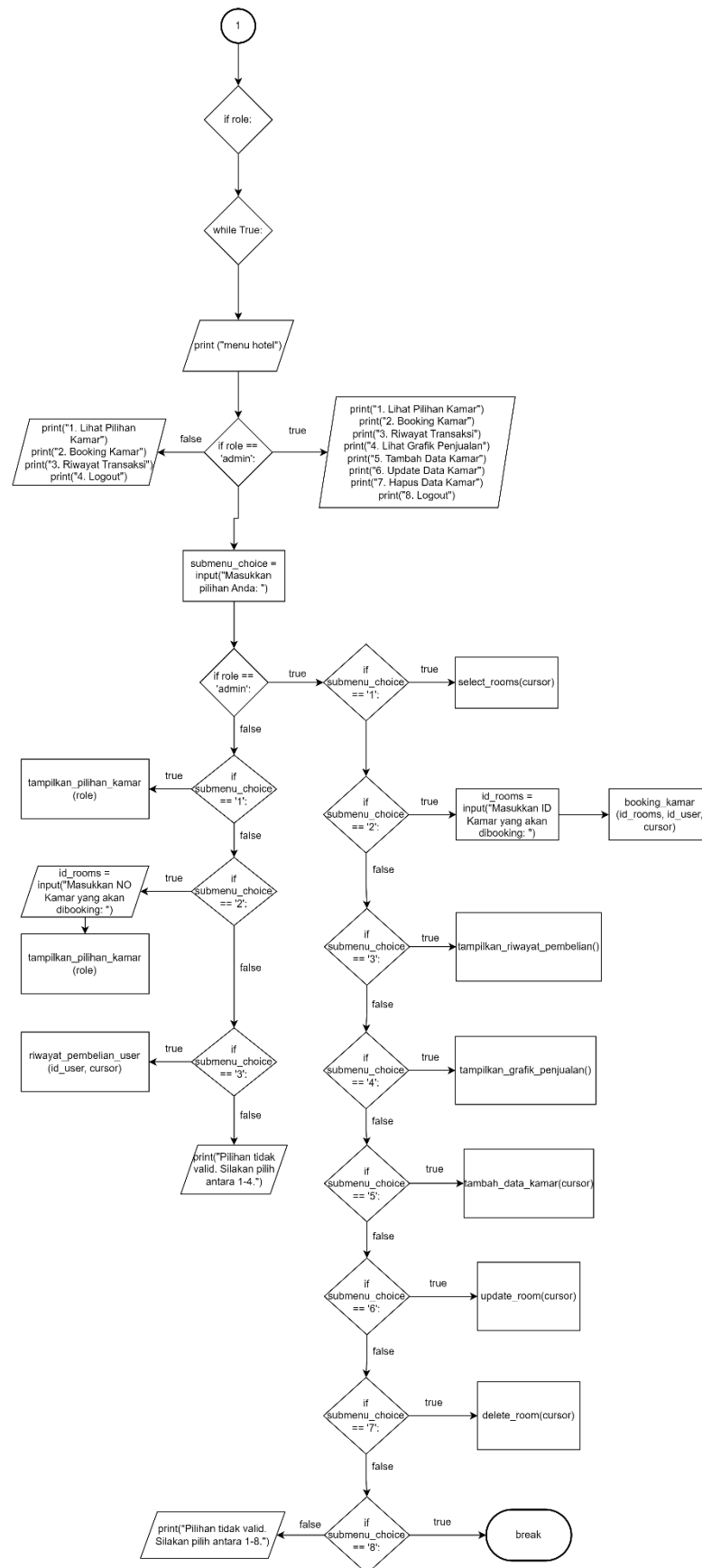
### 3.2.19 Funtion untuk menu login



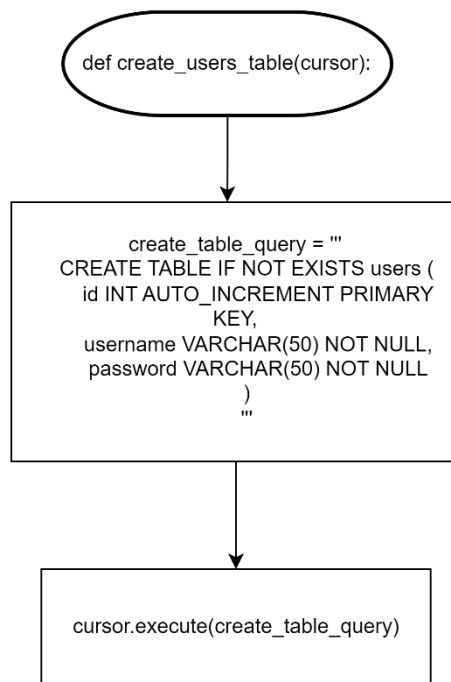
### 3.2.20 Function untuk main program



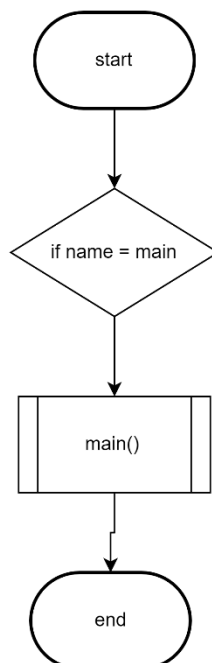
### 3.2.21 Lanjutan function untuk main program



### 3.2.22 Function untuk menambahkan user tidap kali registrasi



### 3.2.23 Program utama



## BAB IV IMPLEMENTASI

### 4.1 Source code

Berikut merupakan source code dari program yang dibuat

#### 4.1.1 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
```

#### 4.1.2 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
```

#### 4.1.3 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)
```

#### 4.1.4 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)
```

#### 4.1.5 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)
```



#### 4.1.6 Code untuk menambahkan riwayat booking

```
def tambahkan_ke_riwayat(action, detail, role):
    riwayat_transaksi.append({'action': action, 'detail': detail, 'role': role})
```

#### 4.1.7 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.")
```

#### 4.1.8 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.")
```

#### 4.1.9 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.")
```

#### 4.1.10 Code untuk validasi password saat registrasi

```
def validate_password(password):  
    # Minimal 8 karakter  
    return len(password) >= 8
```

#### 4.1.11 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
```

#### 4.1.12 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
```

#### 4.1.13 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.")
```

#### 4.1.14 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)
```

#### 4.1.15 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.")
```

#### 4.1.16 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
```

#### 4.1.17 Code untuk close connection

```
def close_connection(connection, cursor=None):
    if cursor:
        cursor.close()
    if connection.is_connected():
        connection.close()
    print("Connection closed")
```

#### 4.1.18 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!")
```

#### 4.1.19 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
```

#### 4.1.20 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.")
```

#### 4.1.21 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.")
```

[illegible]

```

        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 diboeking: ")
            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 diboeking: ")
            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)

```

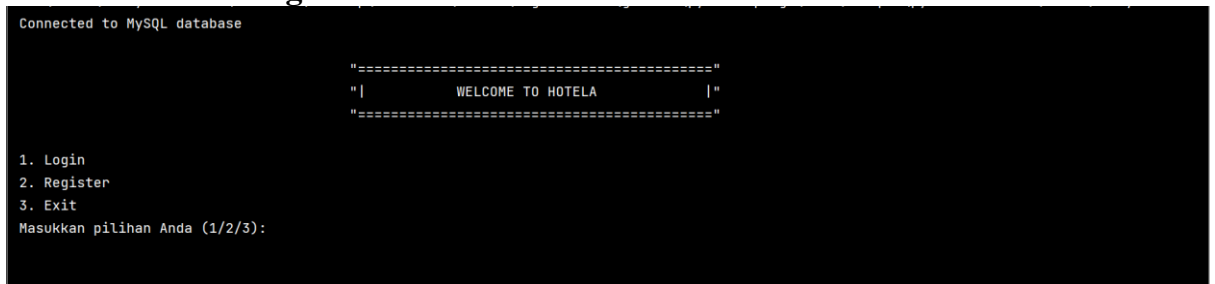
#### 4.1.25 Code untuk program utama

```

if __name__ == "__main__":
    main()

```

## 4.2 Screenshoot Hasil Program



```

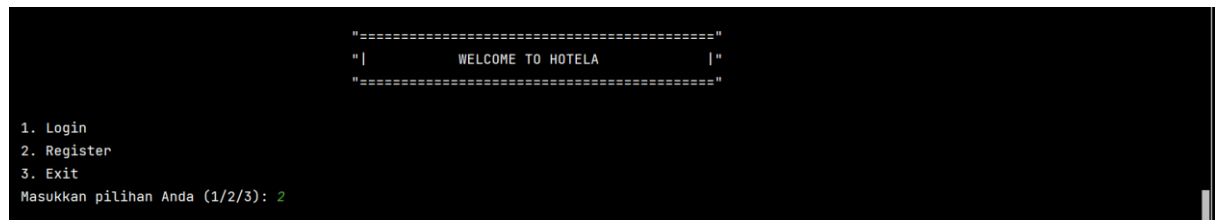
Connected to MySQL database

=====
"|      WELCOME TO HOTELA      |"
=====

1. Login
2. Register
3. Exit
Masukkan pilihan Anda (1/2/3):

```

Gambar 4.2.1 menu login



```

=====
"|      WELCOME TO HOTELA      |"
=====

1. Login
2. Register
3. Exit
Masukkan pilihan Anda (1/2/3): 2

```

Gambar 4.2.2 jika register



```

=====
"|      WELCOME TO HOTELA LOGIN    |"
=====

1. Register User
2. Exit Registration
Masukkan pilihan Anda (1/2): 1
Enter new username: suka
Enter new password: suka
Invalid password. Password must be at least 8 characters long.
Enter new password:
Invalid password. Password must be at least 8 characters long.
Enter new password: Sukasuka.123
Registration successful!

```

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

```

=====
"|          WELCOME TO HOTELA          |"
=====

1. Login
2. Register
3. Exit
Masukkan pilihan Anda (1/2/3): 1

=====
"|          WELCOME TO HOTELA LOGIN      |"
=====

Masukkan username: suka
Masukkan password: Sukasuka.123
Login successful!

```

Gambar 4.2.4 user memilih login

```

=====
"|          WELCOME TO HOTELA MENU      |"
=====

1. Lihat Pilihan Kamar
2. Booking Kamar
3. Riwayat Transaksi
4. Logout
Masukkan pilihan Anda: 1

-----
ID Kamar      | Tipe           | Harga      | Lantai      | Ketersediaan
-----
301           | superior       | 600000     | 3           | booked
401           | deluxe         | 700000     | 4           | booked
502           | twins bed      | 660000     | 5           | booked
702           | deluxe         | 770000     | 7           | booked
703           | deluxe         | 777000     | 7           | booked
704           | deluxe         | 777000     | 7           | booked
705           | deluxe         | 700000     | 7           | Availble
801           | suite          | 1000000    | 8           | booked

```

Gambar 4.2.5 tampilan menu user

```

=====
"|          WELCOME TO HOTELA MENU      |"
=====

1. Lihat Pilihan Kamar
2. Booking Kamar
3. Riwayat Transaksi
4. Logout
Masukkan pilihan Anda: 2

Masukkan NO Kamar yang akan dibooking: 705
id_rooms: 705
kamar_tersedia: (705, 'deluxe', 700000, 7, 'Avaible')
Masukkan Nama: lili
Masukkan Nomor Telepon: 087672838
Masukkan Cara Bayar:cash
Masukkan Tanggal Check-in (YYYY-MM-DD): 2021-01-10
Masukkan Tanggal Check-out (YYYY-MM-DD): 2021-01-11
Total biaya adalah: 700000
Booking successful!

```

Gambar 4.2.6 saat booking berhasil



```
702          | deluxe          | 770000      | 7          | booked
703          | deluxe          | 777000      | 7          | booked
704          | deluxe          | 777000      | 7          | booked
705          | deluxe          | 700000      | 7          | Availble
801          | suite           | 1000000     | 8          | booked

"=====
"|      WELCOME TO HOTELA MENU      |"
"=====

1. Lihat Pilihan Kamar
2. Booking Kamar
3. Riwayat Transaksi
4. Logout
Masukkan pilihan Anda: 2

Masukkan NO Kamar yang akan diboeking: 801
id_rooms: 801
kamar_tersedia: None
Kamar tidak tersedia atau ID Kamar tidak valid.
```

Gambar 4.2.7 saat room yang dipilih tidak tersedia

```
"=====
"|      WELCOME TO HOTELA MENU      |"
"=====

1. Lihat Pilihan Kamar
2. Booking Kamar
3. Riwayat Transaksi
4. Logout
Masukkan pilihan Anda: 3

-----
ID Booking   | Cara Bayar   | Total Bayar | NO Telp      | Nama Tamu   | NO Kamar
-----
131          | cash         | 700000      | 087672838   | lili         | 705
```

Gambar 4.2.8 riwayat transaksi user

```
"=====
"|      WELCOME TO HOTELA MENU      |"
"=====

1. Lihat Pilihan Kamar
2. Booking Kamar
3. Riwayat Transaksi
4. Logout
Masukkan pilihan Anda: 4

"=====
"|      WELCOME TO HOTELA          |"
"=====

1. Login
2. Register
3. Exit
Masukkan pilihan Anda (1/2/3):
```

Gambar 4.2.9 saat user logout

```

=====
|           WELCOME TO HOTELA           |
=====

1. Login
2. Register
3. Exit
Masukkan pilihan Anda (1/2/3): 1

=====
|           WELCOME TO HOTELA LOGIN       |
=====

Masukkan username: pipi
Masukkan password: silvy
Selamat datang, Admin
=====
```

gambar 4.2.10 login admin

```

=====
|           WELCOME TO HOTELA MENU       |
=====

1. Lihat Pilihan Kamar
2. Booking Kamar
3. Riwayat Transaksi
4. Lihat Grafik Penjualan
5. Tambah Data Kamar
6. Update Data Kamar
7. Hapus Data Kamar
8. Logout
Masukkan pilihan Anda:
```

Gambar 4.2.11 menu adminn

```

1. Lihat Pilihan Kamar
2. Booking Kamar
3. Riwayat Transaksi
4. Lihat Grafik Penjualan
5. Tambah Data Kamar
6. Update Data Kamar
7. Hapus Data Kamar
8. Logout
Masukkan pilihan Anda: 1

-----
ID Booking   | NO Kamar   | Type Kamar   | Harga   | Lantai
-----
301          | superior   | 600000       | 3       | booked
401          | deluxe     | 700000       | 4       | booked
502          | twins bed  | 660000       | 5       | booked
702          | deluxe     | 770000       | 7       | booked
703          | deluxe     | 777000       | 7       | booked
704          | deluxe     | 777000       | 7       | booked
705          | deluxe     | 700000       | 7       | booked
801          | suite      | 1000000      | 8       | booked
-----
```

gambar 4.2.12 tampilan riwayat transaksi saat user sudah transaksi

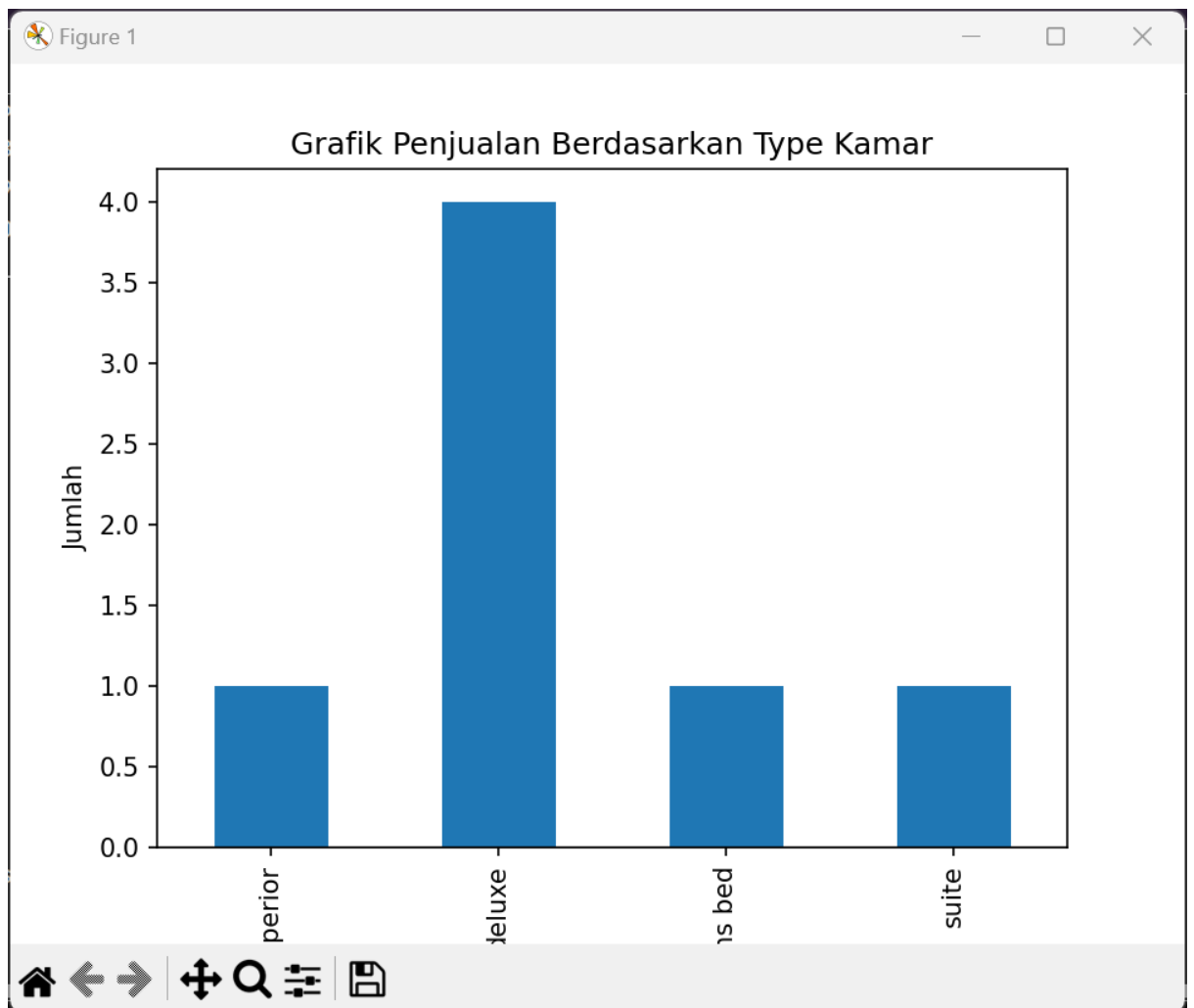
```

1. Lihat Pilihan Kamar
2. Booking Kamar
3. Riwayat Transaksi
4. Lihat Grafik Penjualan
5. Tambah Data Kamar
6. Update Data Kamar
7. Hapus Data Kamar
8. Logout
Masukkan pilihan Anda: 3

```

ID Booking	Cara Bayar	Total Bayar	NO Telp	ID Tamu	Nama Tamu	NO Kamar
123	m-banking	1400000	098786766	12	sel	401
125	cash	770000	0876543456	13	Pipi	401
126	cash	1100000	0876547	14	Pipi	301
128	cash	1540000	09876554	15	pipi	702
129	cash	1320000	08765457	16	maamh	502
130	cash	2000000	087777	17	silvy	801
131	cash	700000	087672838	-	lili	705

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.