

LAPORAN PRAKTIKUM
POSTTEST (7)
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:
Ghina Cyntia Ramadhani (2509106054)
Kelas (B '25)

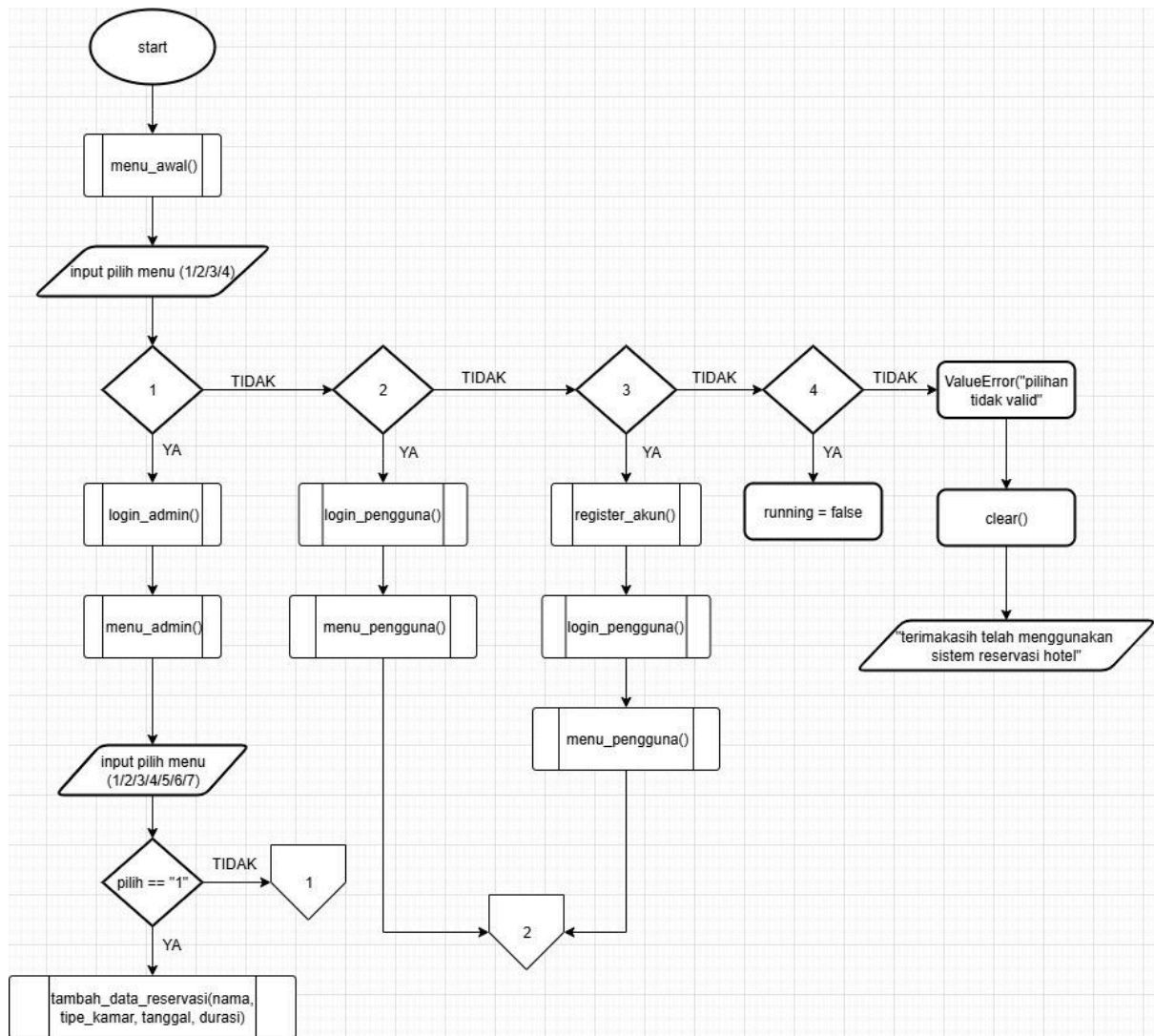
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

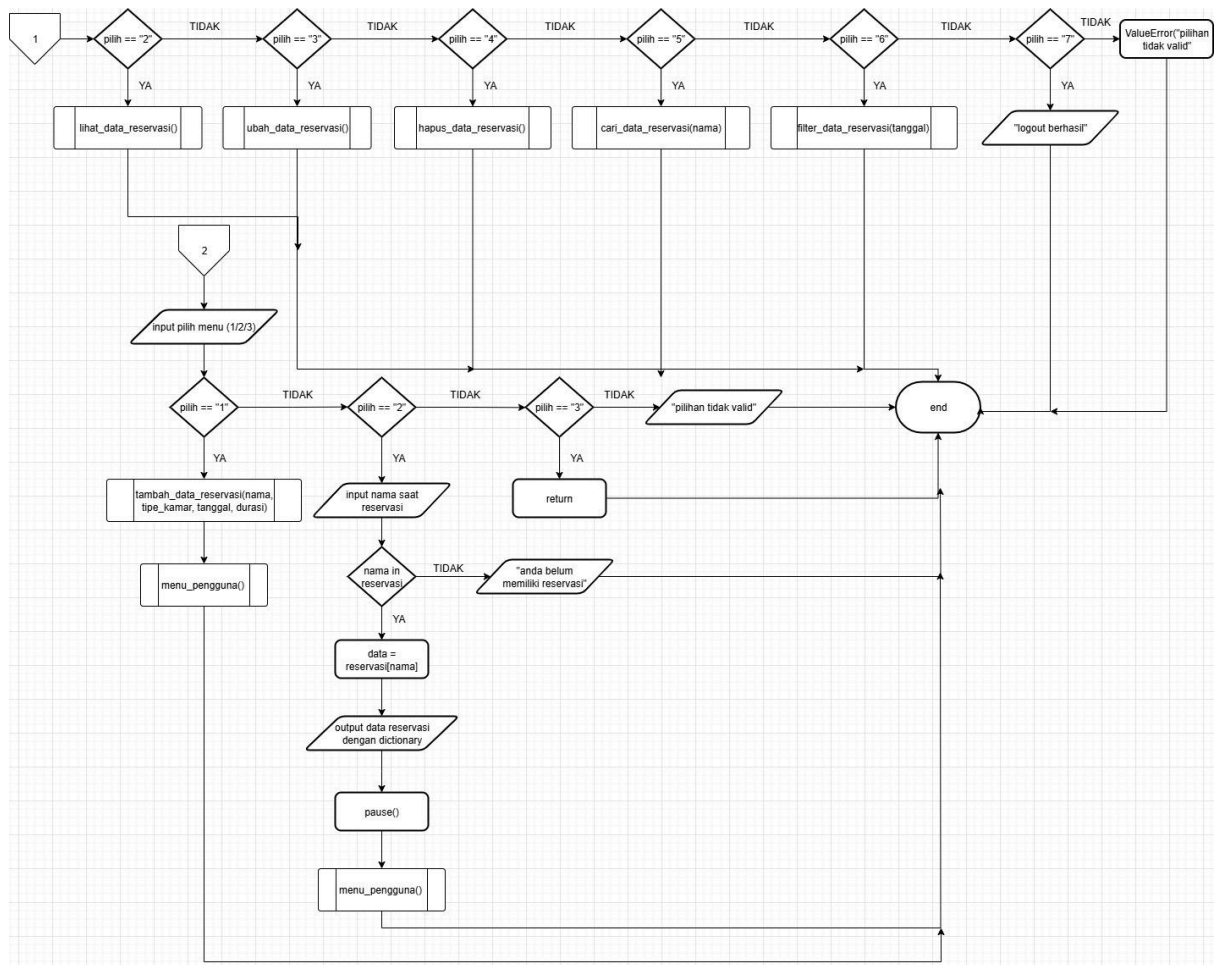
1. Flowchart

Program ini adalah program reservasi hotel yang menggunakan dictionary untuk menyimpan data akun dan data reservasi. Program memiliki dua peran, yaitu admin dan pengguna. Admin dapat melakukan CRUD (Create, Read, Update, dan Delete) terhadap data reservasi, mencari data reservasi, dan memfilter berdasarkan tanggal. Sedangkan, pengguna hanya dapat menambah dan melihat data reservasinya sendiri. Program juga memiliki fitur login, register akun baru, dan logout dengan validasi input agar tidak kosong. Proses login dan register menggunakan dictionary global akun dan data reservasi disimpan di dictionary reservasi.

Program terdiri dari beberapa fungsi yang memudahkan pengaturan alur. Ada fungsi dengan parameter seperti `ubah_data_reservasi(nomor)` yang membutuhkan input agar bisa menjalankan perintah. Dan fungsi tanpa parameter seperti `menu_admin()` yang bisa langsung dijalankan tanpa perlu input tambahan. Fungsi `pause()` dan `clear()` digunakan untuk menghentikan tampilan sementara dan membersihkan layar.

error handling menggunakan Try-except untuk mencegah error saat input atau proses salah. Terdapat juga fungsi rekursif, yang memanggil dirinya sendiri, digunakan pada proses login yang akan mengulang jika input username atau password salah. Secara keseluruhan, Program ini digunakan untuk mengelola data reservasi hotel dengan memanfaatkan konsep fungsi, percabangan, perulangan, dan error handling dalam python.





2. Deskripsi Singkat Program

Tujuan dari program reservasi hotel ini adalah untuk mempermudah proses pengelolaan data pemesanan kamar hotel secara terstruktur dan efisien. Program ini memungkinkan admin untuk menambah, melihat, mengubah, dan menghapus data reservasi, serta mencari data reservasi dengan mudah. Dan dapat mencegah terjadinya double booking. Selain itu, pengguna juga dapat melakukan registrasi akun baru, login, melakukan reservasi kamar sendiri, dan melihat data reservasinya.

Manfaat dari program ini yaitu mempercepat proses reservasi karena semua data tersimpan secara otomatis dalam sistem, mengurangi kesalahan pencatatan karena data tersimpan dalam dictionary, meningkatkan efisiensi admin dalam mengelola banyak reservasi sekaligus, dan memberikan pengalaman pengguna yang praktis.

3. Source Code

Source Code:

```
import os
os.system('cls' if os.name == 'nt' else 'clear')

#data akun & reservasi (variabel global)
akun = {
    "admin": {"username": "admin", "password": "admin123", "role": "admin"},
    "pengguna": {"username": "pengguna", "password": "pengguna123", "role":
"pengguna"}
}
reservasi = {}
running = True

#prosedur
def clear():
    os.system('cls' if os.name == 'nt' else 'clear')

def pause():
```

```

    input("tekan enter untuk melanjutkan...")

#error handling
def tidak_valid(teks):
    if teks.strip() == "":
        raise ValueError("input tidak boleh kosong")

#fungsi tanpa parameter
def menu_awal():
    global running
    clear()
    print("=== SISTEM RESERVASI HOTEL ===")
    print("1. login admin")
    print("2. login pengguna")
    print("3. register")
    print("4. keluar")
    pilih = input("pilih menu (1/2/3/4): ")

    try:
        if pilih == "1":
            login_admin()
        elif pilih == "2":
            login_pengguna()
        elif pilih == "3":
            register_akun()
        elif pilih == "4":
            running = False
        else:
            raise ValueError("pilihan tidak valid")
    except Exception as e:
        print(f"terjadi kesalahan: {e}")
        pause()
        menu_awal()

def menu_admin():
    while True:
        clear()
        print("=== MENU ADMIN ===")
        print("1. tambah data reservasi (create)")
        print("2. lihat data reservasi (read)")
        print("3. ubah data reservasi (update)")

```

```

print("4. hapus data reservasi (delete)")
print("5. cari data reservasi")
print("6. filter berdasarkan tanggal")
print("7. logout")
pilih = input("pilih menu (1/2/3/4/5/6/7): ")

try:
    if pilih == "1":
        clear()
        print("=== TAMBAH DATA RESERVASI ===")
        nama = input("nama: ")
        tipe_kamar = input("tipe kamar (single/superior/deluxe): ")
        tanggal = input("tanggal reservasi (dd-mm-yyyy): ")
        durasi= input("durasi menginap (hari): ")
        tambah_data_reservasi(nama, tipe_kamar, tanggal, durasi)

    elif pilih == "2":
        clear()
        print("=== LIHAT DATA RESERVASI ===")
        lihat_data_reservasi()

    elif pilih == "3":
        clear()
        print("=== UBAH DATA RESERVASI ===")
        ubah_data_reservasi()

    elif pilih == "4":
        clear()
        print("=== HAPUS DATA RESERVASI ===")
        hapus_data_reservasi()

    elif pilih == "5":
        clear()
        print("=== CARI DATA RESERVASI ===")
        nama = input("masukkan nama reservasi yang ingin dicari: ")
        cari_data_reservasi(nama)

    elif pilih == "6":
        clear()
        print("=== FILTER DATA RESERVASI BERDASARKAN TANGGAL ===")
        tanggal = input("masukkan tanggal reservasi yang ingin difilter

```

```

(dd-mm-yyyy): ")
        filter_data_reservasi(tanggal)
    elif pilih == "7":
        print("logout berhasil")
        pause()
        break
    else:
        raise ValueError("pilihan tidak valid")
        pause()
except Exception as e:
    print(f"terjadi kesalahan: {e}")
    pause()

def menu_pengguna():
    global running
    clear()
    print("=== MENU PENGGUNA ===")
    print("1. tambah data reservasi")
    print("2. lihat reservasi saya")
    print("3. logout")
    pilih = input("pilih menu (1/2/3): ")

    try:
        if pilih == "1":
            clear()
            print("=== TAMBAH DATA RESERVASI ===")
            nama = input("nama: ")
            tipe_kamar = input("tipe kamar (single/superior/deluxe): ")
            tanggal = input("tanggal reservasi (dd-mm-yyyy): ")
            durasi = input("durasi menginap (hari): ")
            tambah_data_reservasi(nama, tipe_kamar, tanggal, durasi)
            menu_pengguna()
        elif pilih == "2":
            clear()
            print("=== LIHAT RESERVASI SAYA ===")
            nama = input("masukkan nama yang anda gunakan saat reservasi: ")
            if nama in reservasi:
                data = reservasi[nama]
                print(f"nama: {nama}, tipe kamar: {data['tipe_kamar']}, tanggal
reservasi: {data['tanggal_reservasi']}, durasi menginap: {data['durasi_menginap']}
hari")

```



```

        else:
            print("anda belum memiliki reservasi")
            pause()
            menu_pengguna()
    elif pilih == "3":
        return
    else:
        print("pilihan tidak valid")
        pause()

except Exception as e:
    print(f"terjadi kesalahan: {e}")
    pause()

#fungsi dengan parameter
def tambah_data_reservasi(nama, tipe_kamar, tanggal, durasi):
    clear()
    print("=== TAMBAH DATA RESERVASI ===")
    if cek_ketersediaan_kamar(tipe_kamar, tanggal):
        print("maaf, kamar tidak tersedia pada tanggal tersebut")
        pause()
    else:
        reservasi[nama] = {
            "tipe_kamar": tipe_kamar,
            "tanggal_reservasi": tanggal,
            "durasi_menginap": durasi
        }
        print("data reservasi berhasil ditambahkan")
        pause()

def cek_ketersediaan_kamar(tipe, tanggal):
    for data in reservasi.values():
        if data["tipe_kamar"] == tipe and data["tanggal_reservasi"] == tanggal:
            return True
    return False

def lihat_data_reservasi():
    clear()
    print("=== DATA RESERVASI ===")
    if not reservasi:
        print("tidak ada data reservasi")

```

```

else:
    for i, (nama, data) in enumerate(reservasi.items(), start=1):
        print(f"{i}. nama: {nama}, tipe kamar: {data['tipe_kamar']}, tanggal
reservasi: {data['tanggal_reservasi']}, durasi menginap: {data['durasi_menginap']}
hari")
    pause()

def ubah_data_reservasi():
    clear()
    print("=== UBAH DATA RESERVASI ===")
    if not reservasi:
        print("tidak ada data reservasi untuk diubah")
        pause()
        return

    for i, (nama, data) in enumerate(reservasi.items(), start=1):
        print(f"{i}. nama: {nama}, tipe kamar: {data['tipe_kamar']}, tanggal:
{data['tanggal_reservasi']}, durasi: {data['durasi_menginap']} hari")

    try:
        no = int(input("masukkan nomor data reservasi yang ingin diubah: "))
        if no < 1 or no > len(reservasi):
            print("nomor data reservasi tidak valid")
            pause()
            return

        key_list = list(reservasi.keys())
        key = key_list[no - 1]

        nama_baru = input("masukkan nama baru (kosongkan jika tidak diubah): ") or
key
        tipe_baru = input("masukkan tipe kamar baru (kosongkan jika tidak diubah): ")
or reservasi[key]["tipe_kamar"]
        tanggal_baru = input("masukkan tanggal reservasi baru (kosongkan jika tidak
diubah): ") or reservasi[key]["tanggal_reservasi"]
        durasi_baru = input("masukkan durasi menginap baru (kosongkan jika tidak
diubah): ") or reservasi[key]["durasi_menginap"]

        reservasi.pop(key)
        reservasi[nama_baru] = {
            "tipe_kamar": tipe_baru,

```

```

        "tanggal_reservasi": tanggal_baru,
        "durasi_menginap": durasi_baru
    }
    print("data reservasi berhasil diubah")
    pause()

except Exception as e:
    print(f"terjadi kesalahan: {e}")
    pause()

def hapus_data_reservasi():
    clear()
    print("=== HAPUS DATA RESERVASI ===")
    if not reservasi:
        print("tidak ada data reservasi untuk dihapus")
        pause()
        return
    for i, (nama, data) in enumerate(reservasi.items(), start=1):
        print(f"{i}. nama: {nama}, tipe kamar: {data['tipe_kamar']}, tanggal: {data['tanggal_reservasi']}, durasi: {data['durasi_menginap']} hari")

    try:
        no = int(input("masukkan nomor data reservasi yang ingin dihapus: "))
        if no < 1 or no > len(reservasi):
            print("nomor data reservasi tidak valid")
            pause()
            return

        key_list = list(reservasi.keys())
        key = key_list[no - 1]

        del reservasi[key]
        print("data reservasi berhasil dihapus")
        pause()

    except ValueError as e:
        print(f"terjadi kesalahan: {e}")
        pause()

def cari_data_reservasi(nama):
    clear()
    print("=== CARI DATA RESERVASI ===")

```

```

    if nama in reservasi:
        data = reservasi[nama]
        print(f>Nama: {nama}, Tipe Kamar: {data['tipe_kamar']}, Tanggal Reservasi:
{data['tanggal_reservasi']}, Durasi Menginap: {data['durasi_menginap']} hari")
    else:
        print("data reservasi tidak ditemukan")
    pause()

def filter_data_reservasi(tanggal):
    clear()
    print(f"=== DATA RESERVASI PADA TANGGAL {tanggal} ===")
    ditemukan = False
    for nama, data in reservasi.items():
        if data["tanggal_reservasi"] == tanggal:
            print(f>Nama: {nama}, Tipe Kamar: {data['tipe_kamar']}, Durasi Menginap:
{data['durasi_menginap']} hari")
            ditemukan = True
    if not ditemukan:
        print("tidak ada data reservasi pada tanggal tersebut")
    pause()

#fungsi rekursif
def login_admin():
    clear()
    print("=== LOGIN ADMIN ===")
    try:
        username = input("masukkan username: ")
        password = input("masukkan password: ")

        tidak_valid(username)
        tidak_valid(password)

        if username == akun["admin"]["username"] and password ==
akun["admin"]["password"]:
            print("login berhasil sebagai admin")
            pause()
            menu_admin()
            return
        else:
            print("username atau password salah")
            pause()

```

```

        return login_admin()

except Exception as e:
    print(f"terjadi kesalahan: {e}")
    pause()
    return login_admin()

def login_pengguna():
    clear()
    print("=== LOGIN PENGGUNA ===")
    try:
        username = input("masukkan username: ")
        password = input("masukkan password: ")

        tidak_valid(username)
        tidak_valid(password)

        if username in akun and akun[username]["password"] == password:
            print("login berhasil sebagai pengguna")
            pause()
            menu_pengguna()
            return
        else:
            print("username atau password salah")
            pause()
            return login_pengguna()

    except Exception as e:
        print(f"terjadi kesalahan: {e}")
        pause()
        return login_pengguna()

def register_akun():
    clear()
    global akun
    print("=== REGISTER AKUN ===")
    try:
        username = input("masukkan username baru: ")
        password = input("masukkan password baru: ")

        tidak_valid(username)

```

```

        tidak_valid(password)

    if username in akun:
        print("username sudah terdaftar")
    else:
        akun[username] = {
            "username": username,
            "password": password,
            "role": "pengguna"
        }
        print("registrasi berhasil, silakan login.")
except Exception as e:
    print(f"terjadi kesalahan: {e}")
    pause()
    return register_akun()

#program utama
while running:
    menu_awal()

clear()
print("terima kasih telah menggunakan sistem reservasi hotel")

print("4. keluar")
pilih = input("silahkan pilih menu (1/2/3/4): ")

# LOGIN ADMIN
if pilih == "1":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== LOGIN ADMIN ===")
    username = input("masukkan username: ")
    password = input("masukkan password: ")
    if username in akun and akun[username]["password"] == password:
        role = akun[username]["role"]
    else:
        print("Username atau password salah!")
        input("Tekan ENTER untuk kembali ke menu awal...")
        continue
    if role == "admin":
        while True:
            os.system('cls' if os.name == 'nt' else 'clear')

```

```

print("=== MENU ADMIN ===")
print("1. tambah data reservasi (create)")
print("2. lihat data reservasi (read)")
print("3. ubah data reservasi (update)")
print("4. hapus data reservasi (delete)")
print("5. cari data reservasi")
print("6. filter berdasarkan tanggal")
print("7. logout")
pilihan_admin = input("pilih menu (1/2/3/4/5/6/7): ")

# CREATE
if pilihan_admin == "1":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== TAMBAH DATA RESERVASI ===")
    nama = input("masukkan nama: ")
    tipe = input("masukkan tipe kamar (single/superior/deluxe): ")
    tanggal = input("masukkan tanggal reservasi (dd/mm/yyyy): ")
    durasi = input("masukkan durasi menginap (dalam hari): ")

    # Cek kamar sudah dibooking
    sudah_dibooking = False
    for data in reservasi.values():
        if data["Tipe_Kamar"] == tipe and data["Tanggal_Reservasi"]
== tanggal:
            sudah_dibooking = True
            break

    if sudah_dibooking:
        print("maaf, kamar sudah dibooking pada tanggal tersebut")
        input("tekan enter untuk melanjutkan...")
    else:
        reservasi[nama] = {
            "Tipe_Kamar": tipe,
            "Tanggal_Reservasi": tanggal,
            "Durasi": durasi
        }
        print("data reservasi berhasil ditambahkan")
        input("tekan enter untuk melanjutkan...")

# READ
elif pilihan_admin == "2":

```

```

os.system('cls' if os.name == 'nt' else 'clear')
print("=== DATA RESERVASI ===")
if len(reservasi) == 0:
    print("belum ada data reservasi")
    input("tekan enter untuk melanjutkan...")
else:
    nomor = 1
    for key, data in reservasi.items():
        print(f"{nomor}. Nama: {key}, Tipe Kamar:
{data['Tipe_Kamar']}, Tanggal: {data['Tanggal_Reservasi']}, Durasi: {data['Durasi']}
hari")

        nomor += 1
        input("tekan enter untuk melanjutkan...")

# UPDATE
elif pilihan_admin == "3":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== UBAH DATA RESERVASI ===")
    if len(reservasi) == 0:
        print("belum ada data reservasi")
        input("tekan enter untuk melanjutkan...")
    else:
        nomor = 1
        keys_list = list(reservasi.keys())
        for key in keys_list:
            data = reservasi[key]
            print(f"{nomor}. Nama: {key}, Tipe Kamar:
{data['Tipe_Kamar']}, Tanggal: {data['Tanggal_Reservasi']}, Durasi: {data['Durasi']}
hari")

            nomor += 1
            index_input = input("masukkan nomor data yang ingin diubah:
")

            if index_input.isdigit():
                index = int(index_input) - 1
                if 0 <= index < len(keys_list):
                    key_terpilih = keys_list[index]
                    nama = input("masukkan nama baru (kosongkan jika
tidak diubah): ")

                    kamar = input("masukkan tipe kamar baru (kosongkan
jika tidak diubah): ")

                    tanggal = input("masukkan tanggal reservasi baru

```



```

(kosongkan jika tidak diubah): ")
        durasi = input("masukkan durasi menginap baru
(kosongkan jika tidak diubah): ")
        if nama:
            reservasi[nama]=reservasi.pop(key_terpilih)
            key_terpilih = nama
        if kamar:
            reservasi[key_terpilih]['Tipe_Kamar'] = kamar
        if tanggal:
            reservasi[key_terpilih]['Tanggal_Reservasi'] =
tanggal

        if durasi:
            reservasi[key_terpilih]['Durasi'] = durasi
            print("data reservasi berhasil diubah")
            input("tekan enter untuk melanjutkan...")
        else:
            print("nomor data tidak valid")
            input("tekan enter untuk melanjutkan...")
    else:
        print("input harus angka!")
        input("tekan enter untuk melanjutkan...")

# DELETE
elif pilihan_admin == "4":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== HAPUS DATA RESERVASI ===")
    if len(reservasi) == 0:
        print("belum ada data reservasi")
        input("tekan enter untuk melanjutkan...")
    else:
        nomor = 1
        keys_list = list(reservasi.keys())
        for key in keys_list:
            data = reservasi[key]
            print(f"{nomor}. Nama: {key}, Tipe Kamar:
{data['Tipe_Kamar']}, Tanggal: {data['Tanggal_Reservasi']}, Durasi: {data['Durasi']}
hari")

            nomor += 1
        index_input = input("masukkan nomor data yang ingin dihapus:
")

        if index_input.isdigit():

```

```

        index = int(index_input) - 1
        if 0 <= index < len(keys_list):
            del reservasi[keys_list[index]]
            print("data reservasi berhasil dihapus")
            input("tekan enter untuk melanjutkan...")
        else:
            print("nomor data tidak valid")
            input("tekan enter untuk melanjutkan...")
    else:
        print("input harus angka!")
        input("tekan enter untuk melanjutkan...")

# CARI DATA RESERVASI
elif pilihan_admin == "5":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== CARI DATA RESERVASI ===")
    keyword = input("masukkan nama yang ingin dicari: ")
    hasil = []
    for key, data in reservasi.items():
        if keyword in key:
            hasil.append((key, data))
    if len(hasil) == 0:
        print("data tidak ditemukan")
        input("tekan enter untuk melanjutkan...")
    else:
        for i, (key, data) in enumerate(hasil, 1):
            print(f"{i}. Nama: {key}, Tipe Kamar: {data['Tipe_Kamar']}, Tanggal: {data['Tanggal_Reservasi']}, Durasi: {data['Durasi']} hari")

        input("tekan enter untuk melanjutkan...")

# FILTER BERDASARKAN TANGGAL
elif pilihan_admin == "6":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== FILTER BERDASARKAN TANGGAL ===")
    tanggal_filter = input("masukkan tanggal yang ingin dicari (dd/mm/yyyy): ")

    hasil = []
    for key, data in reservasi.items():
        if data["Tanggal_Reservasi"] == tanggal_filter:
            hasil.append((key, data))

```

```

        if len(hasil) == 0:
            print("tidak ada reservasi pada tanggal tersebut")
            input("tekan enter untuk melanjutkan...")
        else:
            for i, (key, data) in enumerate(hasil, 1):
                print(f"{i}. Nama: {key}, Tipe Kamar:
{data['Tipe_Kamar']], Durasi: {data['Durasi']} hari")
                input("tekan enter untuk melanjutkan...")

# LOGOUT MENU ADMIN
elif pilihan_admin == "7":
    break
else:
    print("pilihan tidak valid")
    input("tekan enter untuk melanjutkan...")

# LOGIN PENGGUNA
elif pilih == "2":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== LOGIN PENGGUNA ===")
    username = input("masukkan username: ")
    password = input("masukkan password: ")

# ERROR HANDLING PENGGUNA
if username == "" or password == "":
    print("Username atau password tidak boleh kosong!")
    input("tekan enter untuk kembali...")
    continue

if username in akun and akun[username]["password"] == password and
akun[username]["role"] == "pengguna":
    print("login berhasil sebagai pengguna")
    input("tekan enter untuk melanjutkan...")

# MENU PENGGUNA
while True:
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== MENU PENGGUNA ===")
    print("1. buat reservasi baru")
    print("2. lihat reservasi saya")
    print("3. logout")

```

```

pilihan_user = input("pilih menu (1/2/3): ")

# CREATE RESERVASI PENGGUNA
if pilihan_user == "1":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== BUAT RESERVASI BARU ===")
    nama = username # nama pemesan = username pengguna
    tipe = input("masukkan tipe kamar (single/superior/deluxe): ")
    tanggal = input("masukkan tanggal reservasi (dd/mm/yyyy): ")
    durasi = input("masukkan durasi menginap (hari): ")

    sudah_dibooking = False
    for data in reservasi.values():
        if data["Tipe_Kamar"] == tipe and data["Tanggal_Reservasi"]
== tanggal:

            sudah_dibooking = True
            break

    if sudah_dibooking:
        print("maaf, kamar sudah dibooking pada tanggal tersebut")
        input("tekan enter untuk melanjutkan...")
    else:
        reservasi[nama] = {
            "Tipe_Kamar": tipe,
            "Tanggal_Reservasi": tanggal,
            "Durasi": durasi
        }
        print("reservasi berhasil dibuat")
        input("tekan enter untuk melanjutkan...")

# READ RESERVASI PENGGUNA
elif pilihan_user == "2":
    os.system('cls' if os.name == 'nt' else 'clear')
    print(f"=== RESERVASI {username} ===")
    if username in reservasi:
        data = reservasi[username]
        print(f>Nama: {username}")
        print(f"Tipe Kamar: {data['Tipe_Kamar']}")
        print(f>Tanggal: {data['Tanggal_Reservasi']}")
        print(f>Durasi: {data['Durasi']} hari")
    else:

```

```

        print("belum ada reservasi")
        input("tekan enter untuk melanjutkan...")

# LOGOUT MENU PENGGUNA
elif pilihan_user == "3":
    break
else:
    print("pilihan tidak valid")
    input("tekan enter untuk melanjutkan...")

# REGISTER
elif pilih == "3":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== REGISTER ===")
    username = input("masukkan username baru: ")
    if username in akun:
        print("username sudah ada!")
        input("tekan enter untuk kembali ke menu utama...")
    else:
        password = input("masukkan password baru: ")
        akun[username] = {"username": username, "password": password, "role":
"pengguna"}
        print("registrasi berhasil")
        input("tekan enter untuk kembali ke menu utama...")

# KELUAR
elif pilih == "4":
    print("Terima kasih telah menggunakan sistem reservasi!")
    break
else:
    print("pilihan tidak valid! masukkan 1/2/3/4")
    input("tekan enter untuk kembali...")

```

4. Hasil Output

```
=== SISTEM RESERVASI HOTEL ===  
1. login admin  
2. login pengguna  
3. register  
4. keluar  
pilih menu (1/2/3/4): 1
```

Gambar 4.1 Screenshot hasil output (menu awal)

```
=== LOGIN ADMIN ===  
masukkan username: admin  
masukkan password: admin123  
login berhasil sebagai admin  
tekan enter untuk melanjutkan...
```

Gambar 4.2 Screenshot hasil output (login admin)

```
=== MENU ADMIN ===  
1. tambah data reservasi (create)  
2. lihat data reservasi (read)  
3. ubah data reservasi (update)  
4. hapus data reservasi (delete)  
5. cari data reservasi  
6. filter berdasarkan tanggal  
7. logout  
pilih menu (1/2/3/4/5/6/7): 1
```

Gambar 4.3 Screenshot hasil output (menu admin)

```
=== TAMBAH DATA RESERVASI ===  
nama: ghina  
tipe kamar (single/superior/deluxe): single  
tanggal reservasi (dd-mm-yyyy): 28-10-2025  
durasi menginap (hari): 1
```

Gambar 4.4 Screenshot hasil output (tambah data reservasi)

```
=== TAMBAH DATA RESERVASI ===  
data reservasi berhasil ditambahkan  
tekan enter untuk melanjutkan...
```

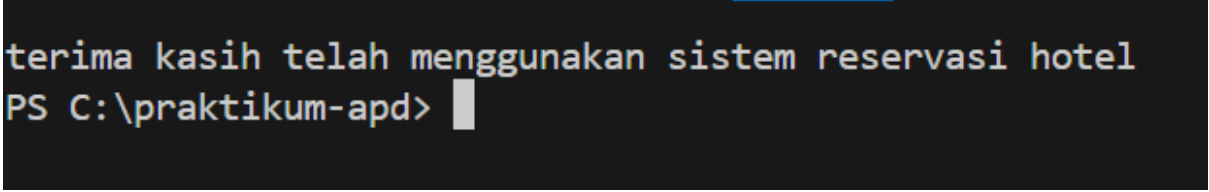
Gambar 4.5 Screenshot hasil output (output berhasil reservasi)

```
=== DATA RESERVASI ===  
1. nama: ghina, tipe kamar: single, tanggal reservasi: 28-10-2025, durasi menginap: 1 hari  
tekan enter untuk melanjutkan...
```

Gambar 4.6 Screenshot hasil output (lihat data reservasi)

```
=== MENU ADMIN ===  
1. tambah data reservasi (create)  
2. lihat data reservasi (read)  
3. ubah data reservasi (update)  
4. hapus data reservasi (delete)  
5. cari data reservasi  
6. filter berdasarkan tanggal  
7. logout  
pilih menu (1/2/3/4/5/6/7): 7  
logout berhasil  
tekan enter untuk melanjutkan...
```

Gambar 4.6 Screenshot hasil output (logout)



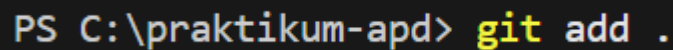
```
terima kasih telah menggunakan sistem reservasi hotel
PS C:\praktikum-apd>
```

Gambar 4.6 Screenshot hasil output (output keluar dari menu awal)

5. Langkah-langkah GIT

5.1 GIT Add

Git add adalah perintah selanjutnya untuk memilih file mana yang akan disimpan perubahannya ke dalam git.

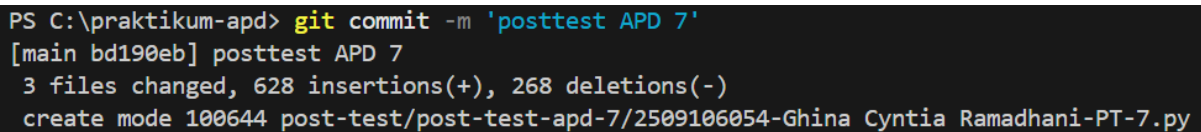


```
PS C:\praktikum-apd> git add .
```

Gambar 5.1 Screenshot GIT Add

5.2 GIT Commit

Git commit adalah perintah untuk menyimpan perubahan file yang sudah dipilih sebelumnya (dengan git add) ke dalam riwayat git.



```
PS C:\praktikum-apd> git commit -m 'posttest APD 7'
[main bd190eb] posttest APD 7
3 files changed, 628 insertions(+), 268 deletions(-)
create mode 100644 post-test/post-test-apd-7/2509106054-Ghina Cyntia Ramadhani-PT-7.py
```

Gambar 5.2 Screenshot GIT Commit

5.3 GIT Push

Git push adalah perintah untuk mengupload perubahan dari repository lokal ke repository online, seperti github.


```
PS C:\praktikum-apd> git push -u origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 5.05 KiB | 739.00 KiB/s, done.
Total 9 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/GhinaCyntia29/praktikum-apd.git
   0820811..bd190eb  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\praktikum-apd>
```

Gambar 5.3 Screenshot GIT Push