

**LAPORAN PRAKTIKUM
POSTTEST (8)
ALGORITMA PEMROGRAMAN DASAR**



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

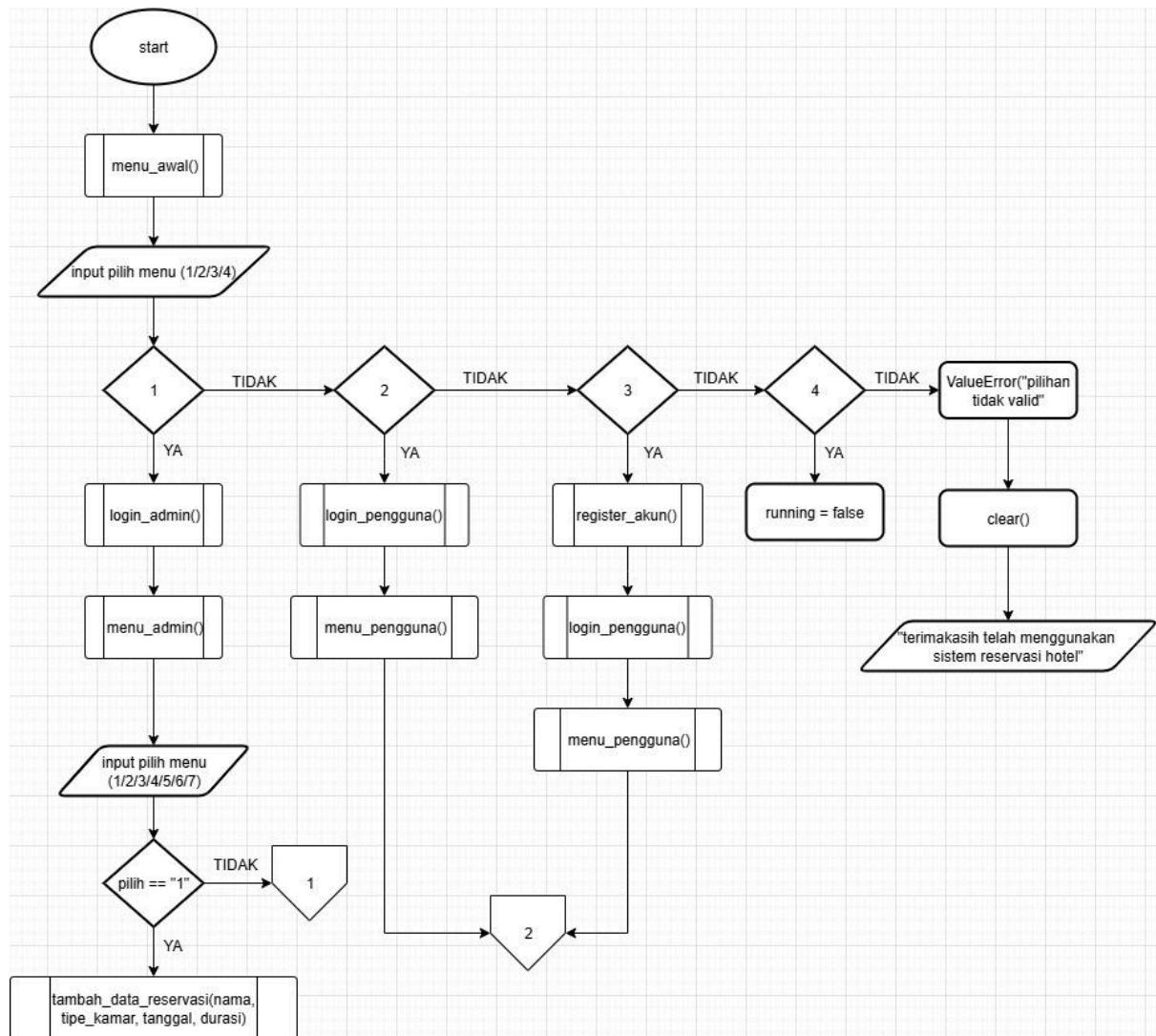
**PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025**

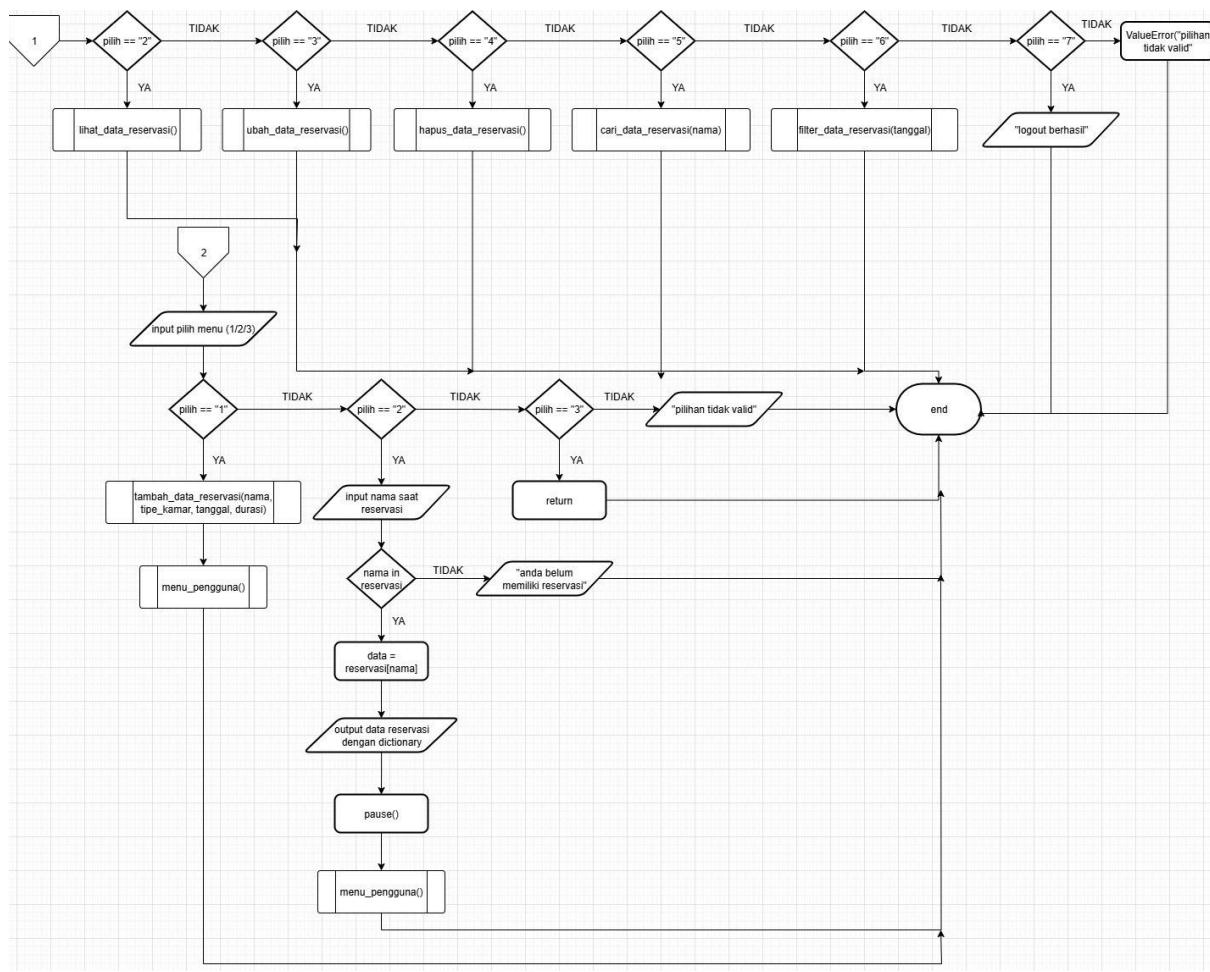
1. Flowchart

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. Semua output data yang muncul ditampilkan dalam bentuk tabel dengan rapi menggunakan library PrettyTable.

Agar pengelolaan kode lebih rapi dan tersrtuktur, program ini dipecah menjadi beberapa file python (modul) yang terpisah sesuai dengan fungsionalitasnya. Modul [main.py](#) berfungsi sebagai pusat alur program dan menampilkan menu utama. Modul [login.py](#) mengatur proses login dan registrasi akun. Modul [reservasi.py](#) berisi fitur CRUD dan filter untuk data reservasi serta menu admin dan menu pengguna. Modul [tampil.py](#) mengatur tampilan data reservasi dalam bentuk tabel menggunakan PrettyTable. Modul [tools.py](#) berisi fungsi pendukung seperti clear() untuk membersihkan layar dan pause() untuk memberi jeda pada program serta error handling. Semua variabel penting seperti data akun dan data reservasi disimpan dalam modul main_data.py, sehingga bisa diakses oleh semua file lain.

Program akan terus berjalan selama variabel running bernilai True, dan akan berakhir ketika pengguna memilih opsi keluar dari program.





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:

```
#tools.py

import os

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

def pause():
    input("tekan enter untuk melanjutkan...")

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

```

#main_data.py

akun = {
    "admin": {"username": "admin", "password": "admin123", "role": "admin"},
    "pengguna": {"username": "pengguna", "password": "pengguna123", "role": "pengguna"}
}

reservasi = {}
running = True


#login.py

from tools import clear, pause, tidak_valid
from main_data import akun
from reservasi import menu_admin, menu_pengguna

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

```

#reservasi.py

```

from tools import clear, pause
from main_data import reservasi
from tampil import tampilkan_reservasi

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 ==")
tampilkan_reservasi(reservasi)
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("== DATA RESERVASI PADA TANGGAL {} ==".format(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()

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()

```

```

#tampil.py

from prettytable import PrettyTable

def tampilkan_reservasi(reservasi):
    if not reservasi:
        print("tidak ada data reservasi")
        return

    table = PrettyTable()
    table.field_names = [ "No", "Nama", "Tipe Kamar", "Tanggal Reservasi", "Durasi Menginap"]

    for i, (nama, data) in enumerate(reservasi.items(), start=1):
        table.add_row([i, nama, data["tipe_kamar"], data["tanggal_reservasi"],
data["durasi_menginap"]])

    print(table)

```

[#main.py](#)

```

from tools import clear, pause, tidak_valid
from login import login_admin, login_pengguna, register_akun
from main_data import akun, reservasi, running
from reservasi import tambah_data_reservasi, lihat_data_reservasi,
ubah_data_reservasi, hapus_data_reservasi, cari_data_reservasi,
filter_data_reservasi

running = True

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:
        print("pilihan tidak valid")
except Exception as e:
    print(f"terjadi kesalahan: {e}")

if __name__ == "__main__":
    while running:
        menu_awal()

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

```

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): 02-11-2025
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 ====
+-----+-----+-----+
| No | Nama | Tipe Kamar | Tanggal Reservasi | Durasi Menginap |
+-----+-----+-----+
| 1 | ghina | single | 02-11-2025 | 1 |
+-----+-----+-----+
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 8'
[main 29a79d5] posttest APD 8
12 files changed, 367 insertions(+)
create mode 100644 post-test/2509106054_Ghina Cyntia Ramadhani_PT-8/__pycache__/_login.cpython-313.pyc
create mode 100644 post-test/2509106054_Ghina Cyntia Ramadhani_PT-8/__pycache__/_main_data.cpython-313.pyc
create mode 100644 post-test/2509106054_Ghina Cyntia Ramadhani_PT-8/__pycache__/_reservasi.cpython-313.pyc
create mode 100644 post-test/2509106054_Ghina Cyntia Ramadhani_PT-8/__pycache__/_tampil.cpython-313.pyc
create mode 100644 post-test/2509106054_Ghina Cyntia Ramadhani_PT-8/__pycache__/_tools.cpython-313.pyc
create mode 100644 post-test/2509106054_Ghina Cyntia Ramadhani_PT-8/login.py
create mode 100644 post-test/2509106054_Ghina Cyntia Ramadhani_PT-8/main.py
create mode 100644 post-test/2509106054_Ghina Cyntia Ramadhani_PT-8/main_data.py
create mode 100644 post-test/2509106054_Ghina Cyntia Ramadhani_PT-8/reservasi.py
create mode 100644 post-test/2509106054_Ghina Cyntia Ramadhani_PT-8/tampil.py
create mode 100644 post-test/2509106054_Ghina Cyntia Ramadhani_PT-8/tools.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: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 12 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (18/18), 10.91 KiB | 1.56 MiB/s, done.
Total 18 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/GhinaCyntia29/praktikum-apd.git
  d728c0d..29a79d5  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\praktikum-apd> █
```

Gambar 5.3 Screenshot GIT Push