LAPORAN PRAKTIKUM

**POSTTEST 7**

**ALGORITMA PEMROGRAMAN DASAR**

****

**Disusun oleh:**

**Muhammad Ihsan Rosadi 2509106081**

**Kelas B2‘25**

**PROGRAM STUDI INFORMATIKA**

**UNIVERSITAS MULAWARMAN**

**SAMARINDA**

**2025**

# 1. Flowchart

Gambar 1.1 Flowchart Menu Utama & User Biasa

Gambar 1.2 Flowchart Menu Admin

Penjelasan singkat:

Flowchart diatas menggambarkan alur sistem manajemen proyek Desain Grafis yang memiliki dua peran utama: User Biasa dan Admin.

1. Menu Utama

1. Sistem dimulai dengan menampilkan menu utama kepada pengguna.
2. Pengguna dapat memilih untuk:
   1. Login sebagai Admin
   2. Login sebagai User Biasa
   3. Keluar dari sistem

Jika input pilihan tidak valid, sistem menampilkan pesan “Pilihan Tidak Valid”.

2. Login

1. Pengguna memasukkan Username dan Password.
2. Sistem akan memverifikasi data login:
3. Jika username atau password salah → pesan error ditampilkan.
4. Jika benar → pengguna diarahkan ke menu sesuai peran (Admin/User).

3. Menu Admin

Admin memiliki beberapa opsi utama:

1. Tambah Proyek – Admin memasukkan judul, deskripsi, status, dan data proyek baru.
2. Lihat Proyek – Menampilkan daftar proyek yang ada.
3. Edit Proyek – Admin dapat mengubah data proyek tertentu (judul, deskripsi, software, deadline, client, status, dsb).
4. Hapus Proyek – Menghapus proyek dari sistem.
5. Logout – Keluar dari sistem.

Setiap langkah dilengkapi validasi, misalnya jika proyek tidak ditemukan atau input salah, akan muncul pesan “Pilihan Tidak Valid” atau “Proyek Belum Ada”.

4. Menu User Biasa

User Biasa hanya dapat:

1. Lihat Daftar Proyek yang tersedia.
2. Logout dari sistem.

Jika pilihan salah, sistem memberi pesan “Pilihan Tidak Valid”.

5. Validasi dan Pengulangan

1. Jika terdapat input yang tidak sesuai, sistem akan meminta pengguna untuk mengulang pilihan.
2. Setelah aksi selesai (misalnya menambah atau mengedit proyek), sistem kembali ke menu utama peran masing-masing.

6. End

Flowchart berakhir setelah pengguna memilih keluar dari sistem.

## 2. Deskripsi Singkat Program

Program ini merupakan sistem manajemen proyek desain grafis berbasis terminal yang memungkinkan pengguna melakukan login, registrasi, dan mengelola proyek. Program ini membedakan antara akun admin dan user biasa: admin dapat menambah, melihat, memperbarui, dan menghapus proyek, sedangkan user biasa hanya dapat melihat daftar proyek. Data pengguna dan proyek disimpan dalam daftar Python sederhana (list), dan antarmuka menggunakan menu teks yang bersih dengan perintah os.system('cls'/'clear') untuk membersihkan layar. Program ini berjalan dalam loop utama hingga pengguna memilih untuk keluar.

# 3. Source Code

**A. Inisialisasi & Variable Global**

Menyimpan data pengguna, proyek, dan variabel global untuk pengguna yang sedang login.

| import os  users = [  {"username": "mihsanr81", "password": "2509106081", "role": "admin"} ] projects = [] user\_ditemukan = None |
| --- |

**B. Fungsi Utilitas (Pendukung Tampilan)**

Membersihkan layar dan menunggu input sebelum lanjut, agar tampilan program lebih rapi.

| def clear\_screen():  os.system('cls' if os.name == 'nt' else 'clear')  def pause():  input("Tekan Enter Untuk Melanjutkan...") |
| --- |

**C. Menu Awal Program**

Menampilkan menu utama (Login, Register, Keluar) dan mengembalikan pilihan pengguna.

| def tampilkan\_menu\_awal():  clear\_screen()  print("=== SISTEM MANAJEMEN PROYEK DESAIN GRAFIS ===")  print("1. Login")  print("2. Register")  print("3. Keluar")  return input("Pilih menu: ") |
| --- |

**D. Fitur Tambah Proyek**

Menambahkan proyek baru ke daftar dengan validasi input.

| def tambah\_proyek(username):  global projects  clear\_screen()  print("=== TAMBAH PROYEK ===")  try:  judul = input("Judul: ")  deskripsi = input("Deskripsi: ")  software = input("Software: ")  deadline = input("Deadline: ")  klien = input("Klien: ")  status = input("Status: ")   if judul == "" or klien == "":  raise ValueError("Judul dan Klien wajib diisi!")   project\_baru = {  "judul": judul,  "deskripsi": deskripsi,  "software": software,  "deadline": deadline,  "klien": klien,  "status": status  }  projects.append(project\_baru)  print("Proyek berhasil ditambah!")  except ValueError as e:  print(e)  finally:  pause() |
| --- |

**E. Fitur Lihat Proyek**

Menampilkan semua data proyek yang tersimpan.

| def lihat\_proyek():  clear\_screen()  print("=== DAFTAR PROYEK ===")  if len(projects) == 0:  print("Belum ada proyek.")  else:  for i, p in enumerate(projects):  print(f"\n[{i+1}] {p['judul']}")  print(f"Deskripsi: {p['deskripsi']}")  print(f"Software : {p['software']}")  print(f"Deadline : {p['deadline']}")  print(f"Klien : {p['klien']}")  print(f"Status : {p['status']}")  pause() |
| --- |

**F. Fitur Update Proyek**

Mengubah data proyek berdasarkan nomor indeks yang dipilih.

| def update\_proyek(index):  global projects  try:  if len(projects) == 0:  raise Exception("Belum ada proyek untuk diubah.")  if not (0 <= index < len(projects)):  raise IndexError("Nomor proyek tidak valid!")   while True:  clear\_screen()  print(f"=== UBAH PROYEK: {projects[index]['judul']} ===")  print("1. Judul")  print("2. Deskripsi")  print("3. Software")  print("4. Deadline")  print("5. Klien")  print("6. Status")  print("7. Selesai Update")  pilih = input("Pilih data yang ingin diubah: ")   if pilih == "1":  projects[index]["judul"] = input("Masukkan judul baru: ")  elif pilih == "2":  projects[index]["deskripsi"] = input("Masukkan deskripsi baru: ")  elif pilih == "3":  projects[index]["software"] = input("Masukkan software baru: ")  elif pilih == "4":  projects[index]["deadline"] = input("Masukkan deadline baru: ")  elif pilih == "5":  projects[index]["klien"] = input("Masukkan nama klien baru: ")  elif pilih == "6":  projects[index]["status"] = input("Masukkan status baru: ")  elif pilih == "7":  break  else:  print("Pilihan tidak valid!")  pause()  print("Data proyek berhasil diperbarui!")  except Exception as e:  print(e)  finally:  pause() |
| --- |

**G. Fitur Hapus Proyek**

Menghapus proyek berdasarkan pilihan nomor dari daftar.

| def hapus\_proyek():  global projects  clear\_screen()  print("=== HAPUS PROYEK ===")  try:  if len(projects) == 0:  raise Exception("Belum ada proyek.")  for i, p in enumerate(projects):  print(f"[{i+1}] {p['judul']}")  nomor = input("Pilih nomor proyek: ")  if not nomor.isdigit():  raise ValueError("Input harus angka.")   nomor = int(nomor) - 1  if not (0 <= nomor < len(projects)):  raise IndexError("Nomor tidak valid.")   hapus = input(f"Hapus proyek '{projects[nomor]['judul']}'? (y/n): ").lower()  if hapus == "y":  del projects[nomor]  print("Proyek dihapus!")  else:  print("Dibatalkan.")  except (ValueError, IndexError, Exception) as e:  print(e)  finally:  pause() |
| --- |

**H. Fitur Pilih dan Update Proyek**

Menampilkan daftar proyek dan meminta pengguna memilih proyek yang ingin diubah.

| def pilih\_dan\_update\_proyek():  clear\_screen()  print("=== UBAH PROYEK ===")  if len(projects) == 0:  print("Belum ada proyek.")  pause()  return  for i, p in enumerate(projects):  print(f"[{i+1}] {p['judul']}")  nomor = input("\nPilih nomor proyek: ")  if nomor.isdigit():  update\_proyek(int(nomor) - 1)  else:  print("Input harus angka!")  pause() |
| --- |

**I. Menu Admin**

Menu khusus admin untuk melakukan CRUD proyek dan logout.

| def menu\_admin():  clear\_screen()  print("=== MENU ADMIN ===")  print("1. Tambah Proyek")  print("2. Lihat Proyek")  print("3. Update Proyek")  print("4. Hapus Proyek")  print("5. Logout")   pilih = input("Pilih: ")   if pilih == "1":  tambah\_proyek(user\_ditemukan["username"])  elif pilih == "2":  lihat\_proyek()  elif pilih == "3":  pilih\_dan\_update\_proyek()  elif pilih == "4":  hapus\_proyek()  elif pilih == "5":  return  else:  print("Pilihan tidak valid!")  pause()   menu\_admin() |
| --- |

**J. Menu Pengguna Biasa**

Menu sederhana bagi user biasa untuk melihat proyek dan logout.

| def menu\_pengguna():  while True:  clear\_screen()  print(f"=== MENU PENGGUNA ({user\_ditemukan['username']}) ===")  print("1. Lihat Proyek")  print("2. Logout")  pilih\_user = input("Pilih: ")   if pilih\_user == "1":  lihat\_proyek()  elif pilih\_user == "2":  break  else:  print("Pilihan tidak valid!")  pause() |
| --- |

**K. Fitur Login**

Memverifikasi akun lalu arahkan ke menu admin/pengguna.

| def login():  global user\_ditemukan  clear\_screen()  print("=== LOGIN ===")  username = input("Username: ")  password = input("Password: ")   user\_ditemukan = None  for u in users:  if u["username"] == username and u["password"] == password:  user\_ditemukan = u  break   if user\_ditemukan is None:  print("Login gagal! Username atau password salah.")  pause()  return   if user\_ditemukan["role"] == "admin":  menu\_admin()  else:  menu\_pengguna() |
| --- |

**L. Fitur Register**

Membuat akun baru dengan pengecekan username duplikat.

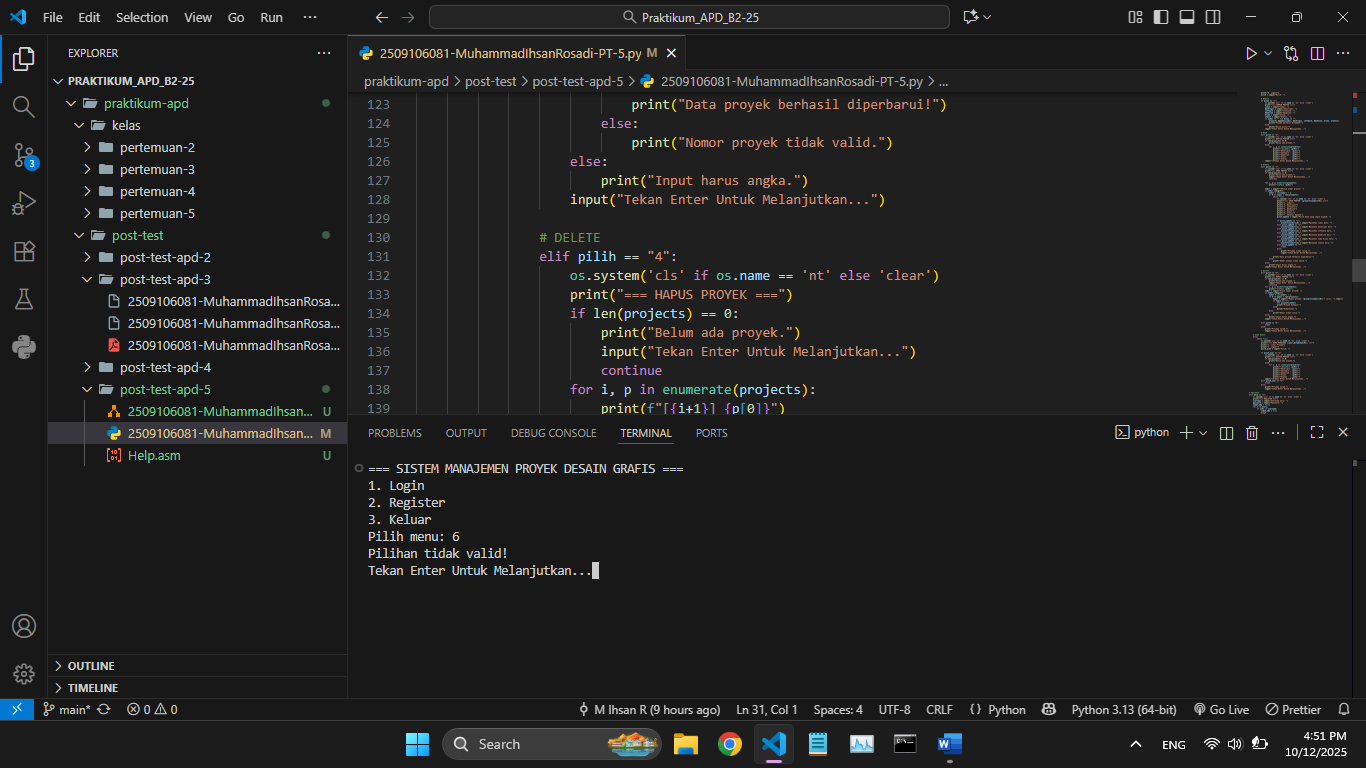
| def register():  clear\_screen()  print("=== REGISTER ===")  username = input("Username baru: ")  password = input("Password: ")  sudah\_ada = any(u["username"] == username for u in users)   if username == "" or password == "":  print("Username dan password wajib diisi!")  elif sudah\_ada:  print("Username sudah digunakan!")  else:  users.append({"username": username, "password": password, "role": "user"})  print("Akun berhasil dibuat!")  pause() |
| --- |

**M. Fungsi Utama Program**

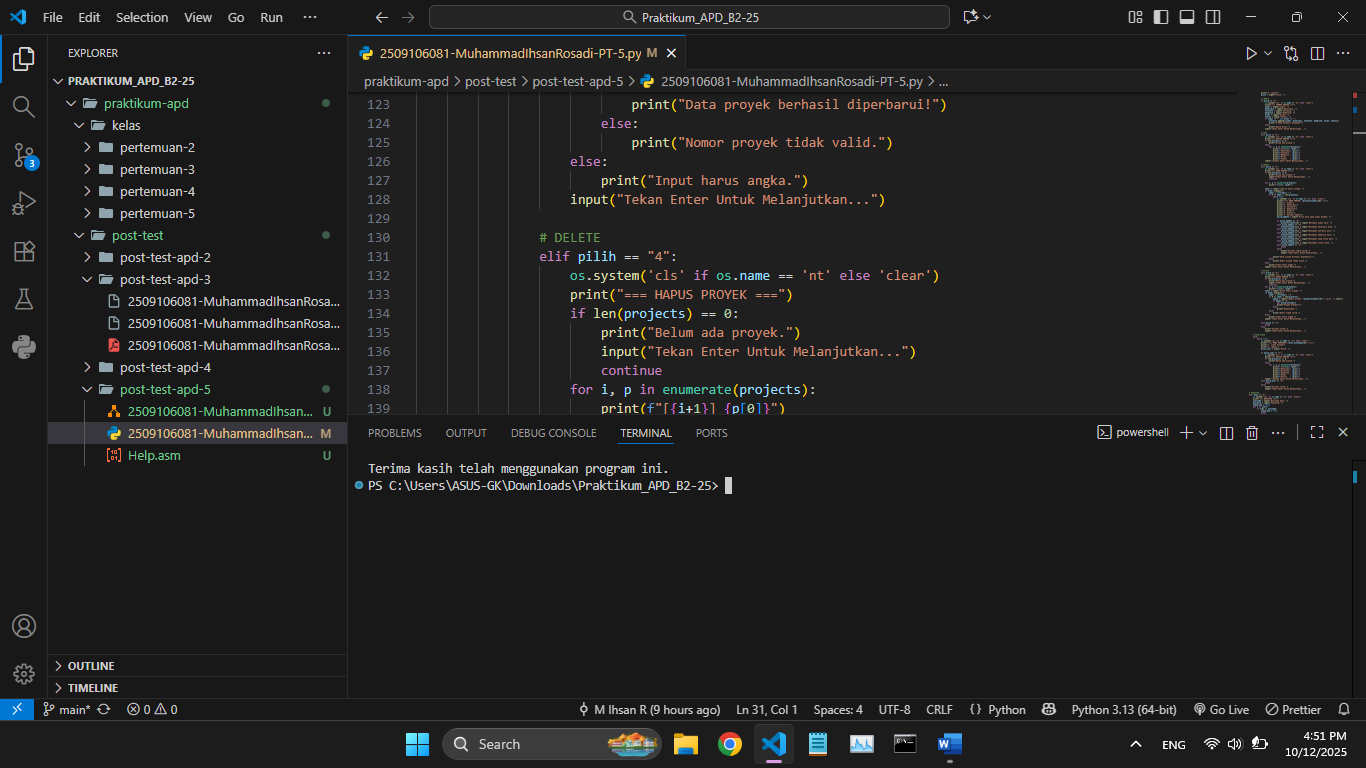
Menjalankan alur utama program, menampilkan menu awal, dan mengatur navigasi antar fitur.

| def main():  while True:  menu = tampilkan\_menu\_awal()  if menu == "1":  login()  elif menu == "2":  register()  elif menu == "3":  clear\_screen()  print("Terima kasih telah menggunakan program ini.")  break  else:  print("Pilihan tidak valid!")  pause()  main() |
| --- |

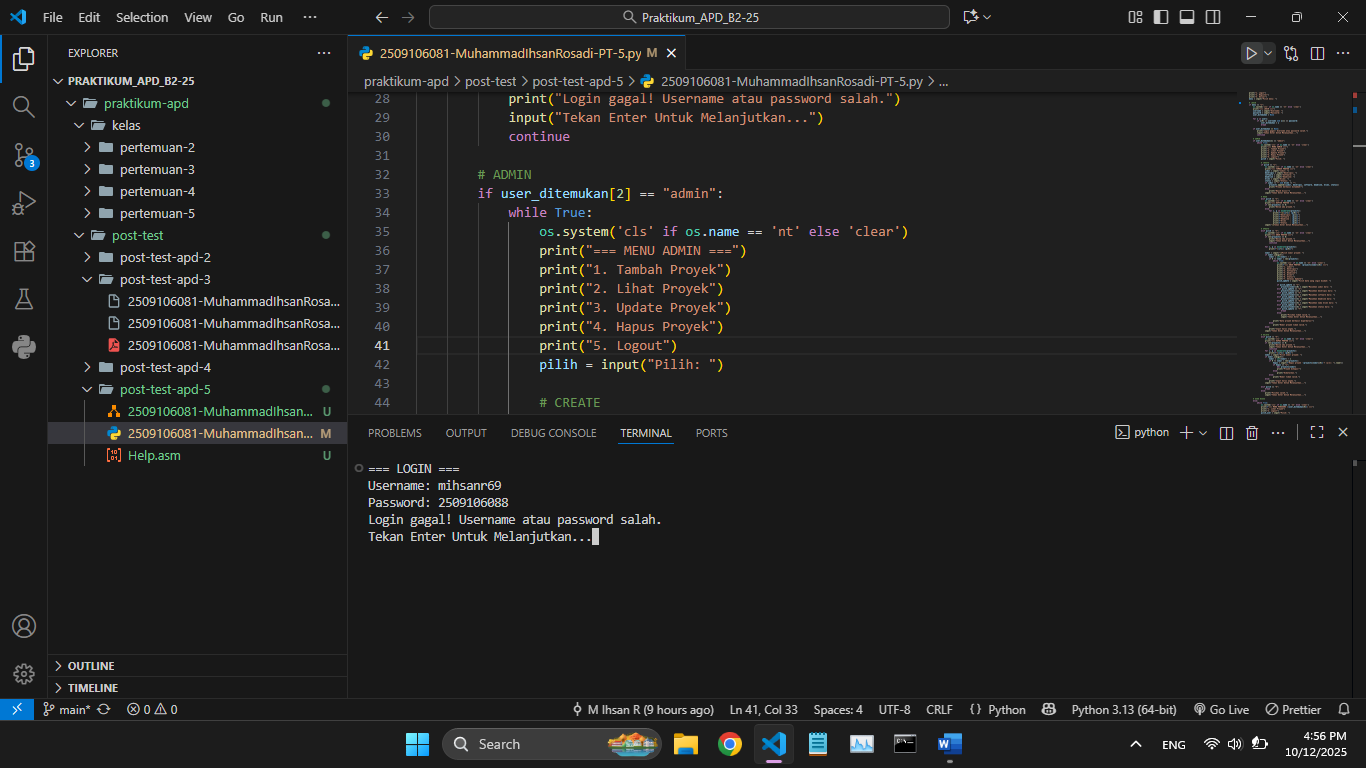
## 4. Hasil Output



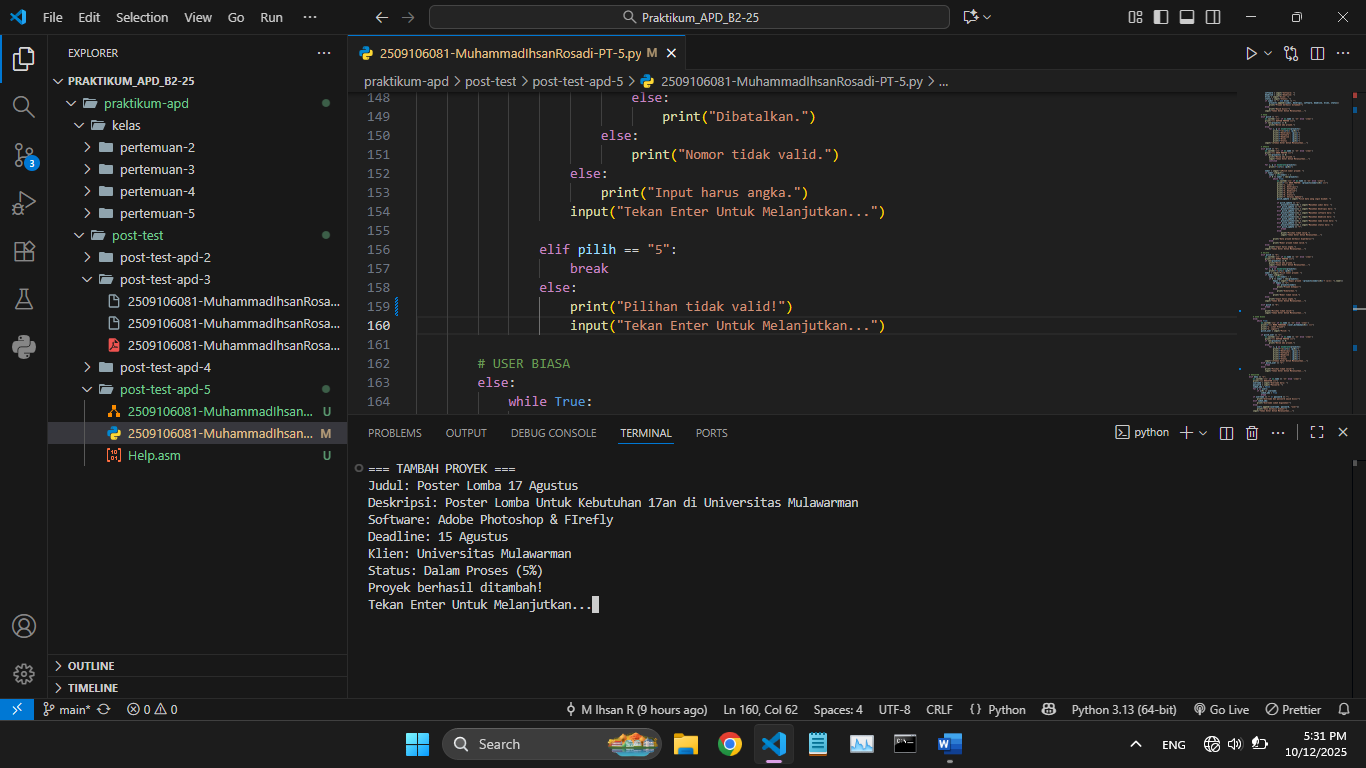
Gambar 4.1 Fitur Menu Utama/Admin/User Biasa (Input Nomor Invalid)



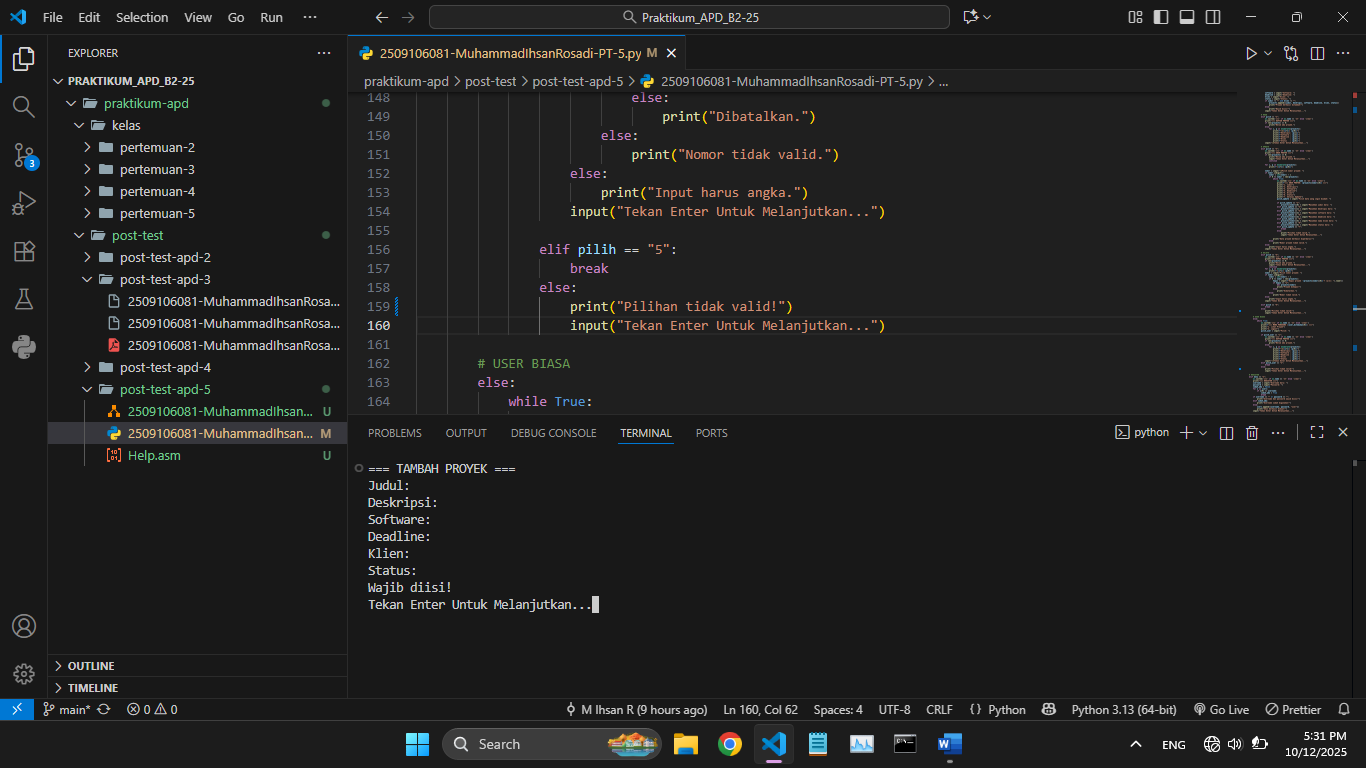
Gambar 4.2 Fitur Keluar



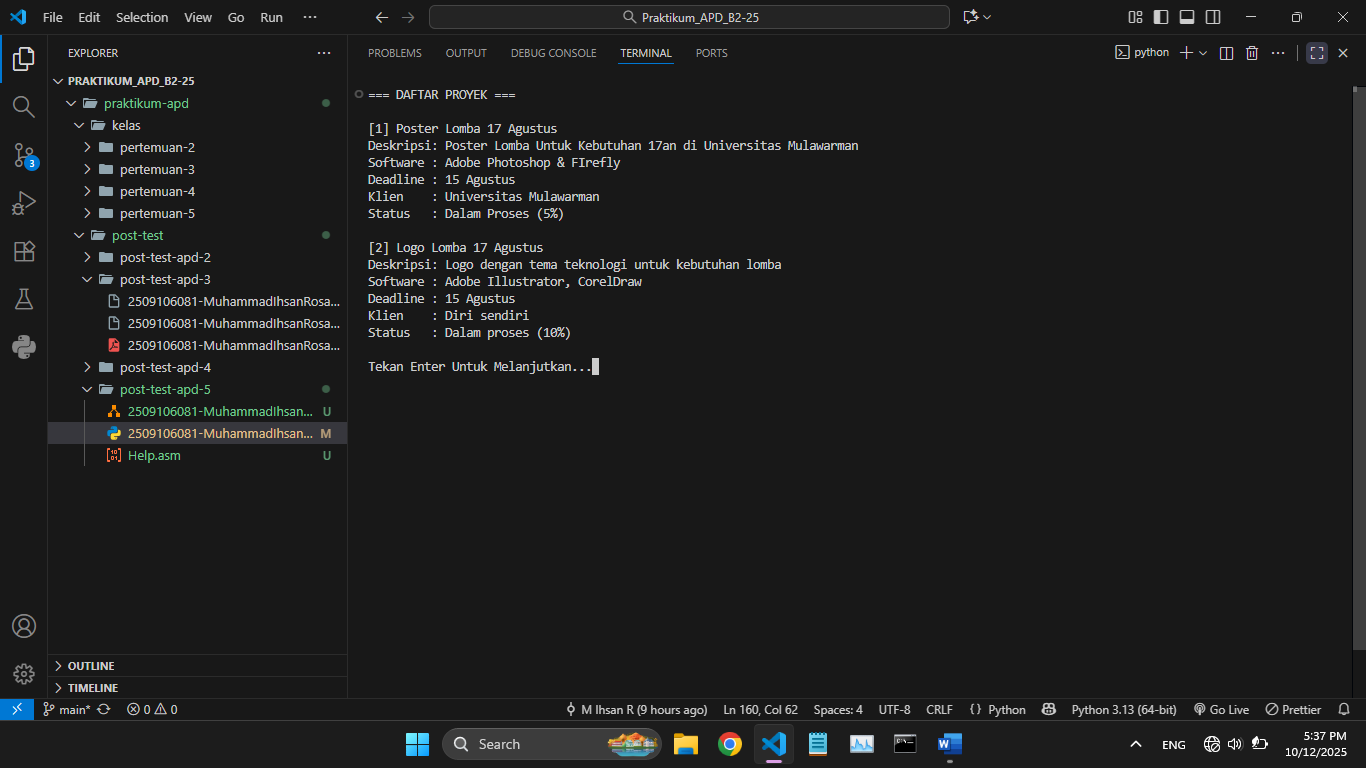
Gambar 4.3 Fitur Login (Gagal)



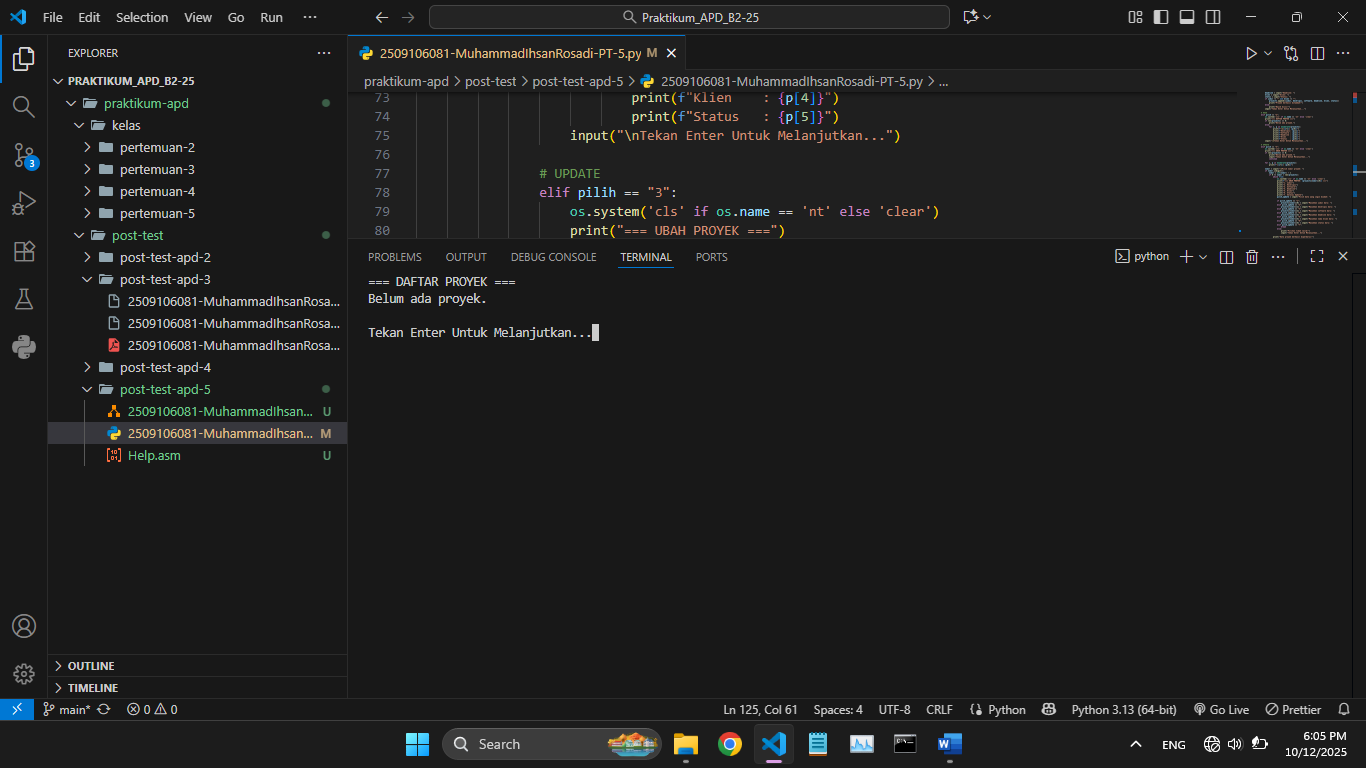
Gambar 4.4 Fitur Create/Menambah Proyek (Berhasil)



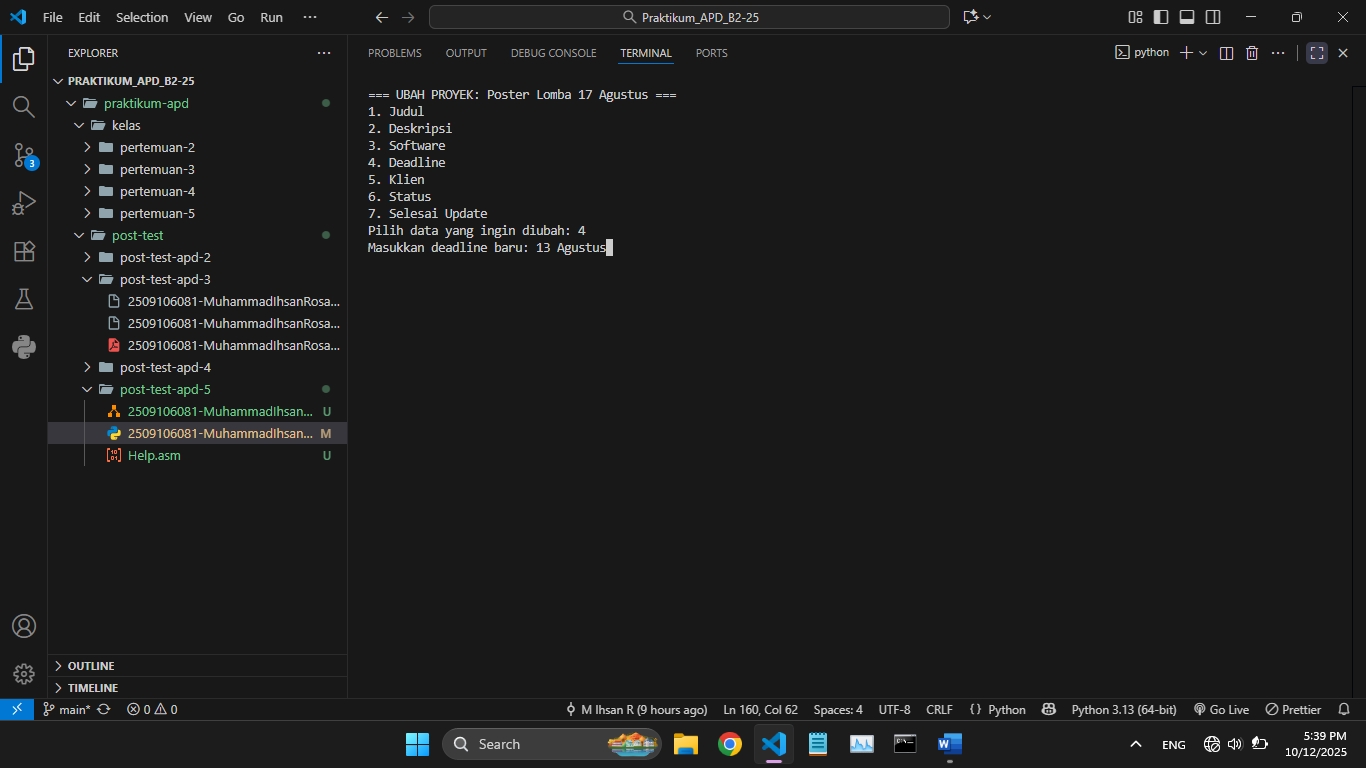
Gambar 4.5 Fitur Create/Menambah Proyek (Gagal)

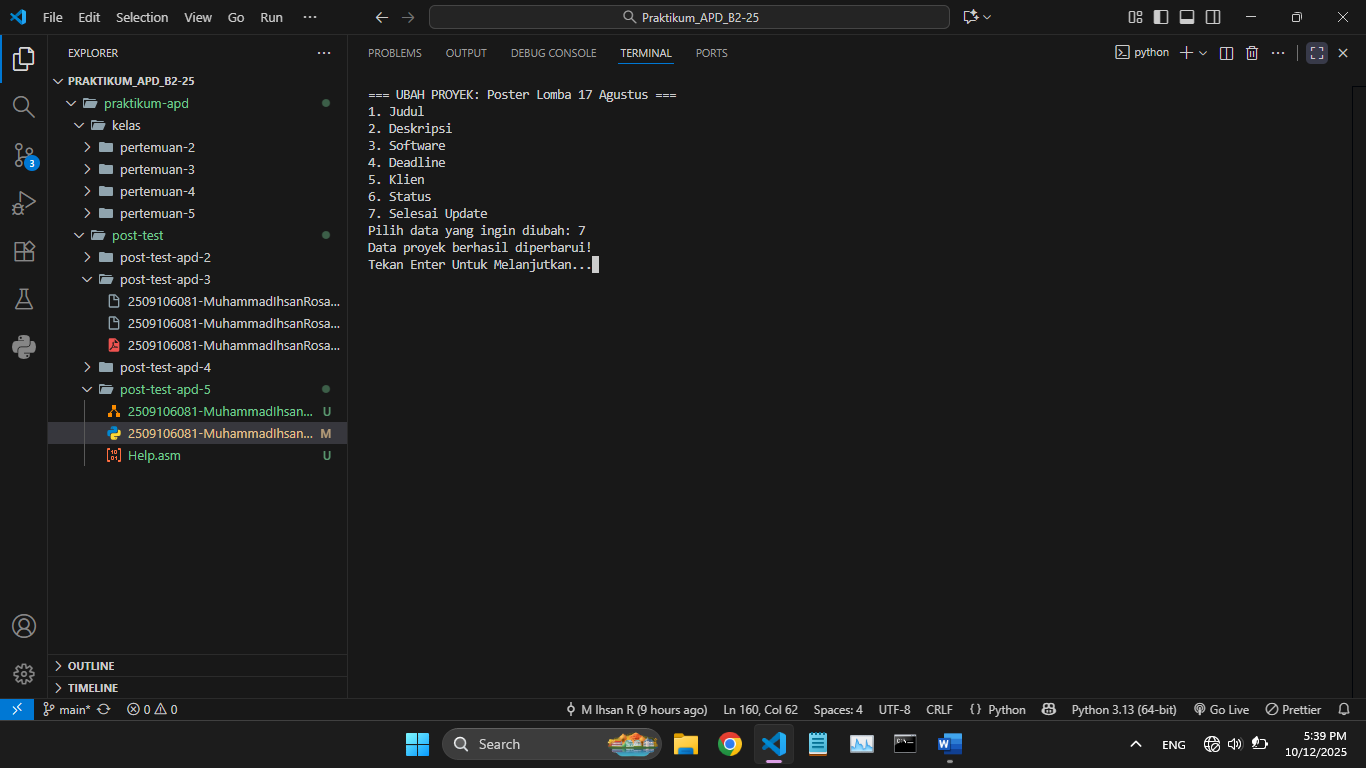


Gambar 4.6 Fitur Read/Melihat Proyek (Ada Proyek)

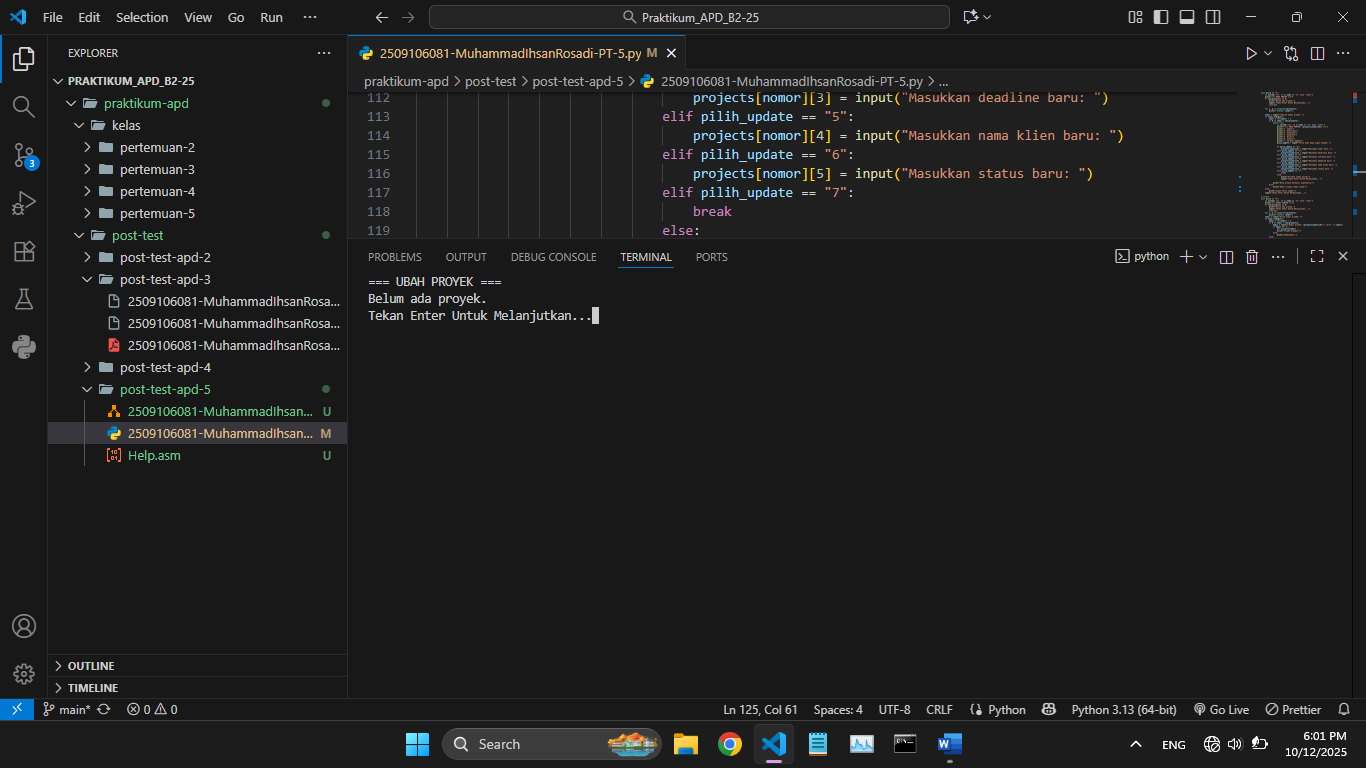


Gambar 4.7 Fitur Read/Melihat Proyek (Tidak Ada Proyek)

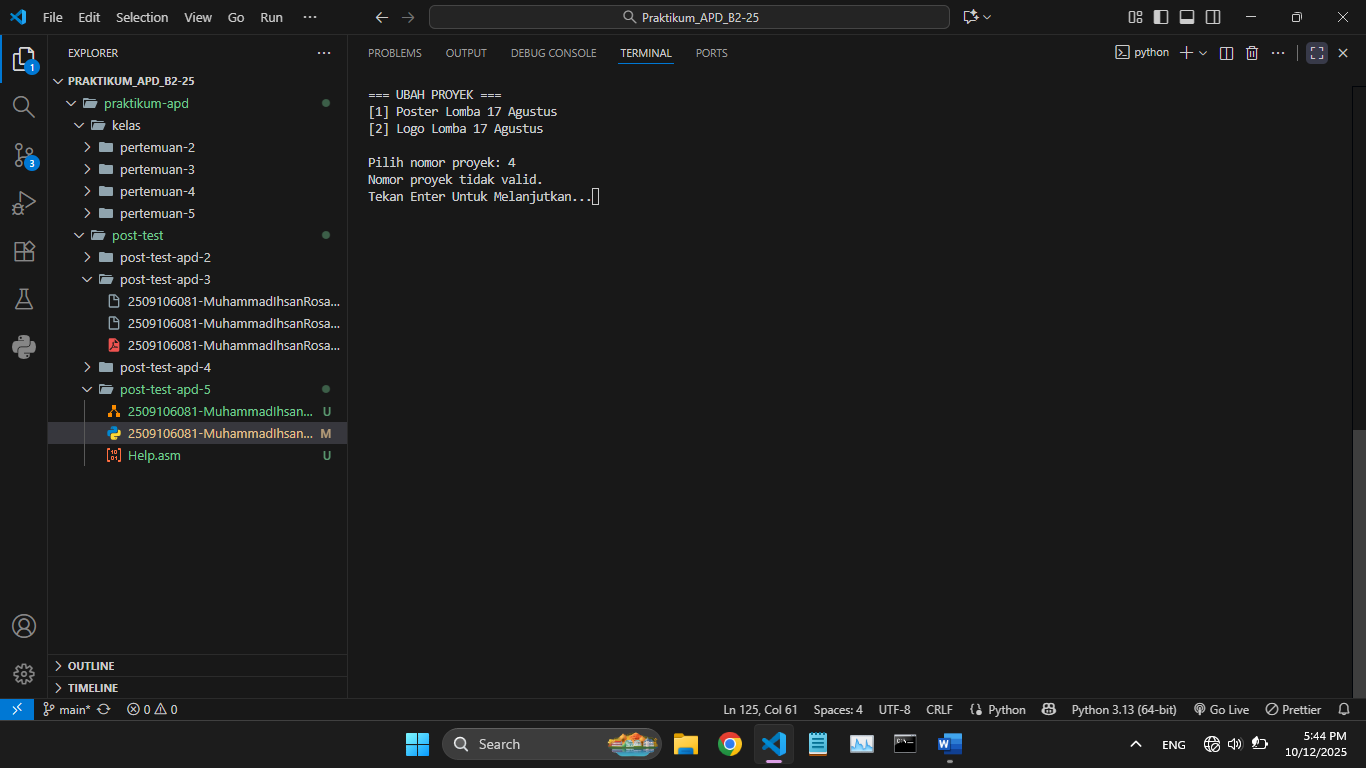




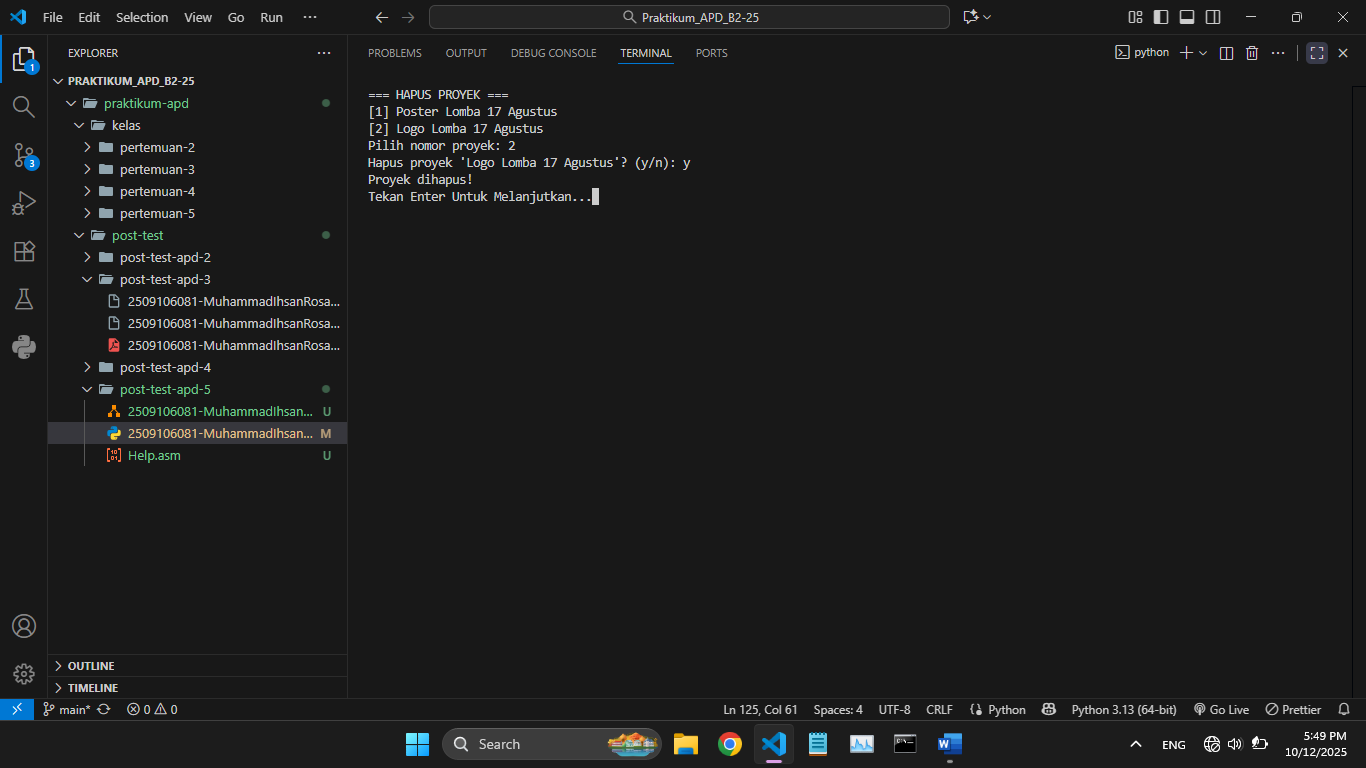
Gambar 4.8 Fitur Update/Memperbarui Proyek (Ada Proyek) 2



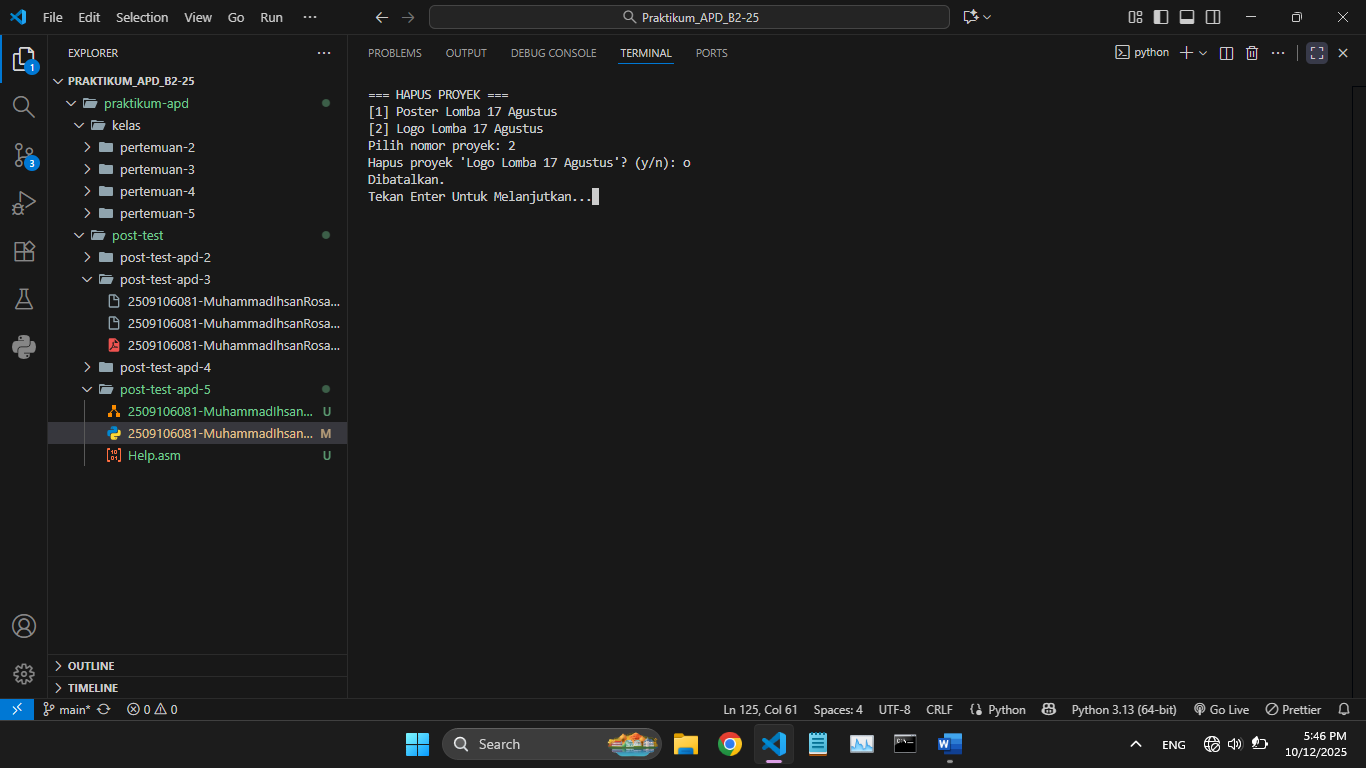
Gambar 4.9 Fitur Update/Memperbarui Proyek (Tidak Ada Proyek)



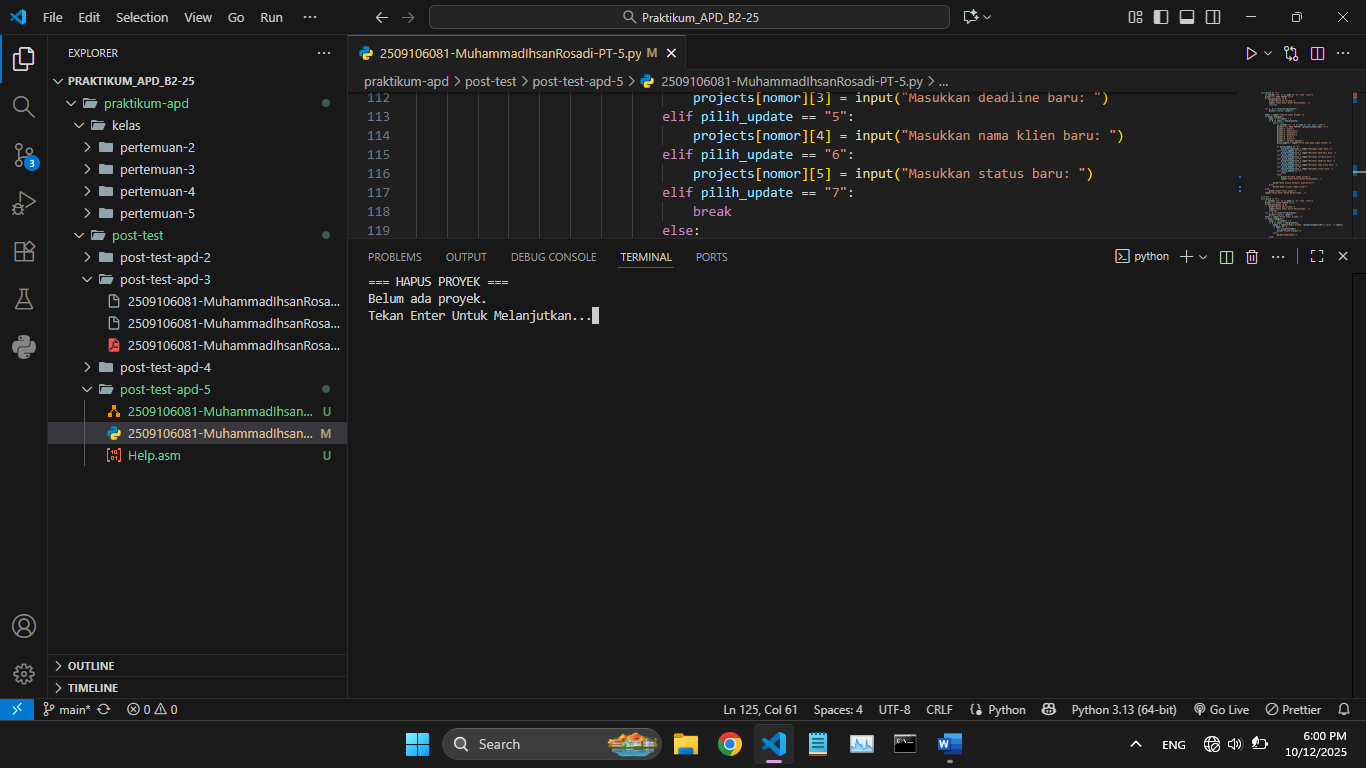
Gambar 4.10 Fitur Update/Memperbarui Proyek (Input Nomor Invalid)



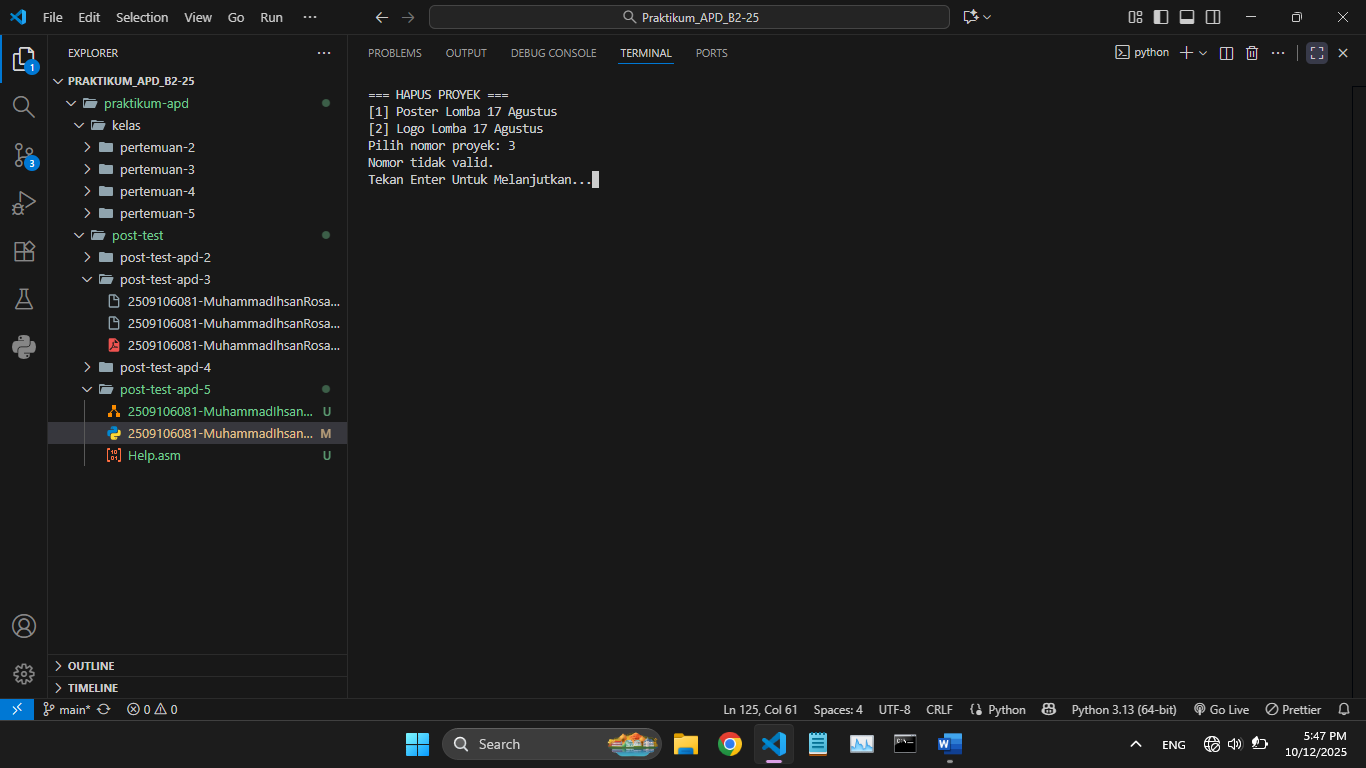
Gambar 4.11 Fitur Delete/Menghapus Proyek (Ada Proyek (Input y))



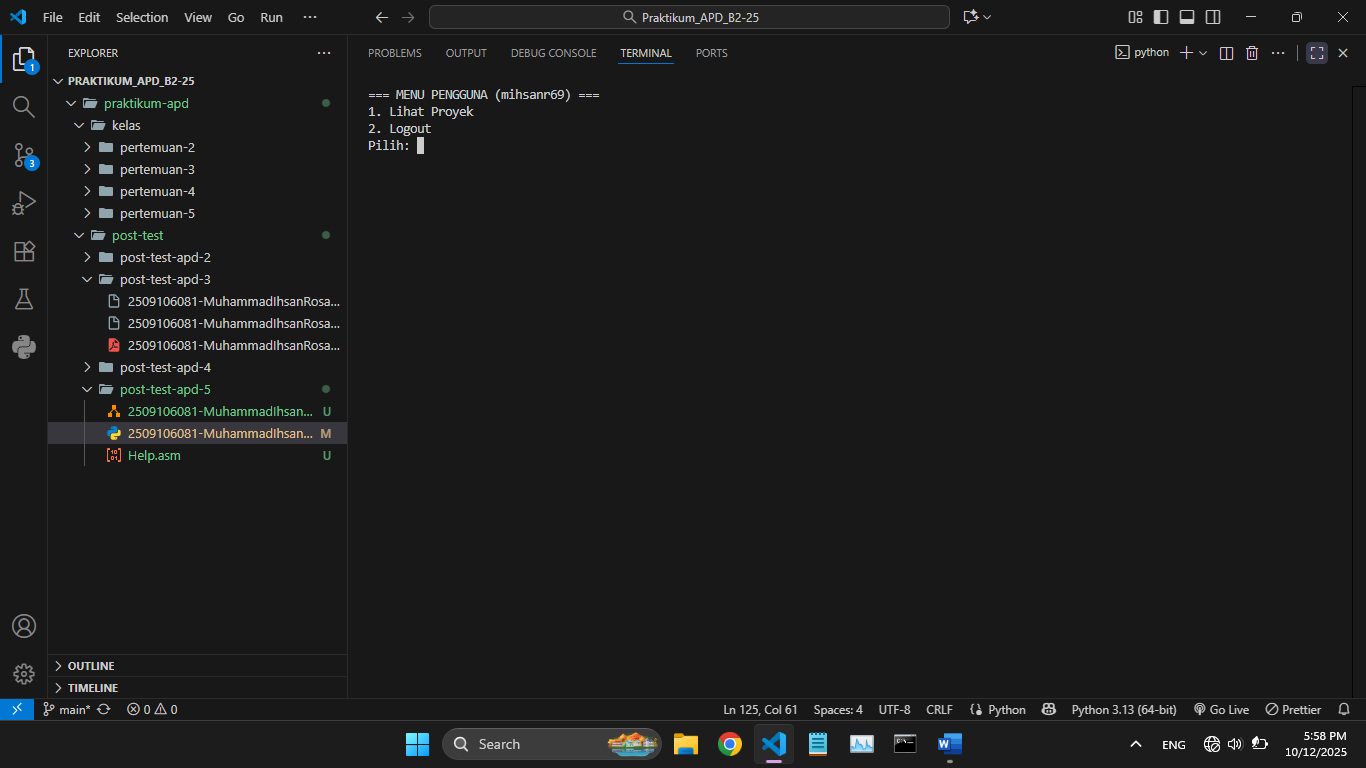
Gambar 4.12 Fitur Delete/Menghapus Proyek (Ada Proyek (Input Selain y))



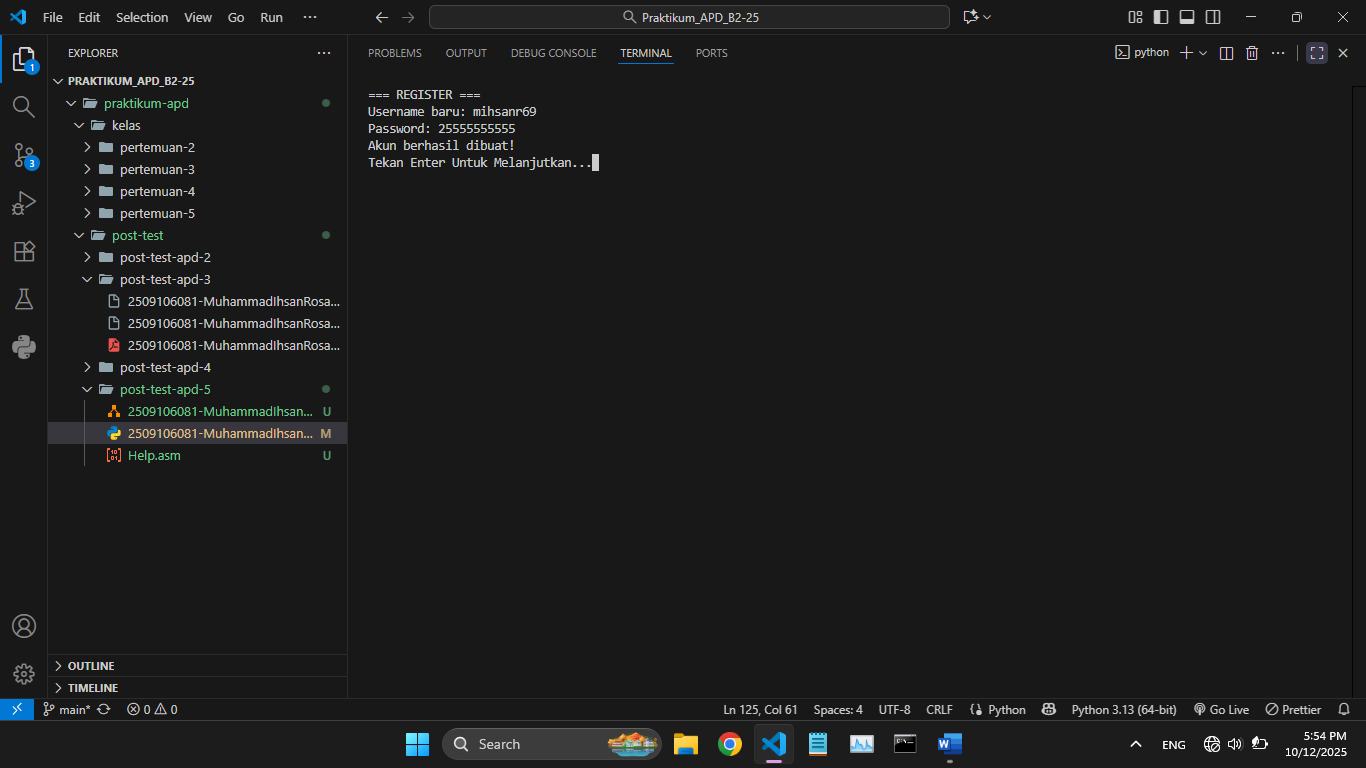
Gambar 4.13 Fitur Delete/Menghapus Proyek (Tidak Ada Proyek)



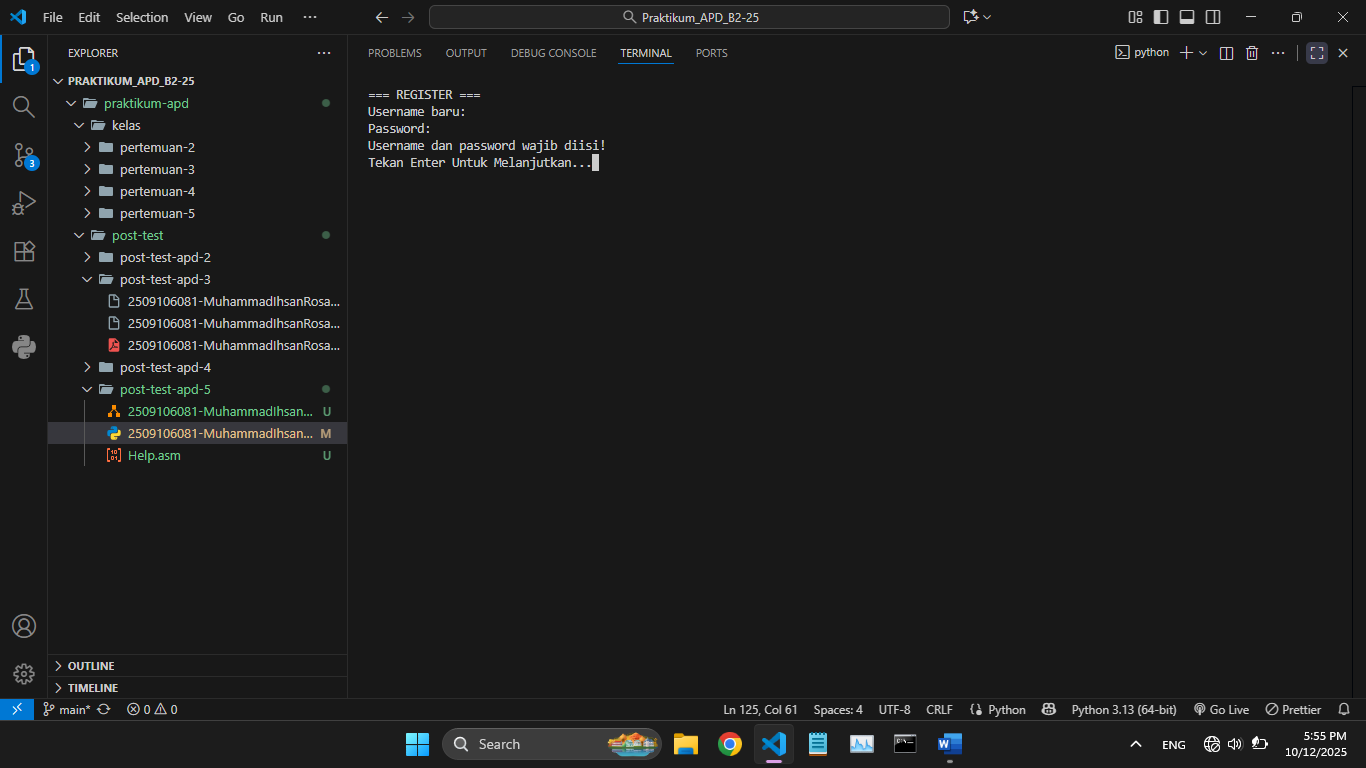
Gambar 4.14 Fitur Delete/Menghapus Proyek (Input Nomor Invalid)



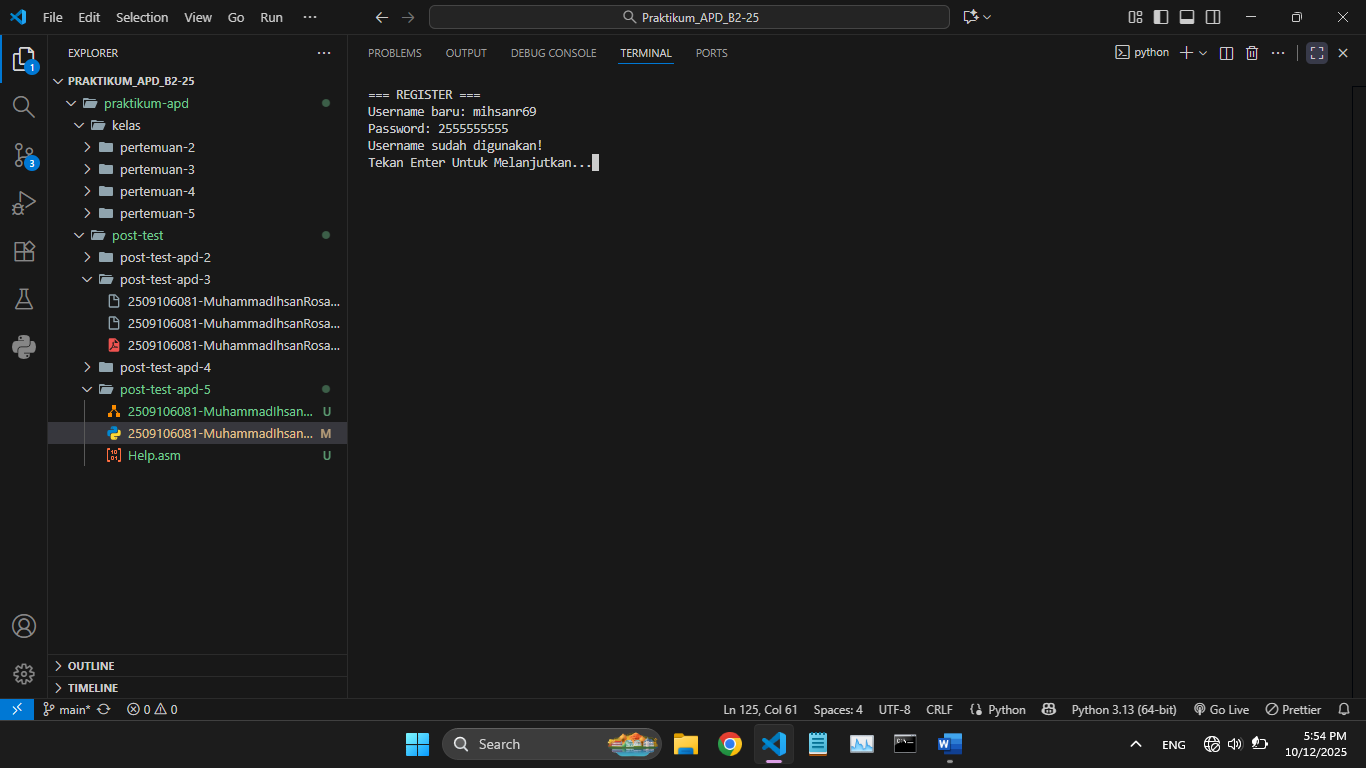
Gambar 4.15 Fitur Menu User Biasa



Gambar 4.16 Fitur Register (Berhasil)



Gambar 4.16 Fitur Register (Gagal/Kosong)

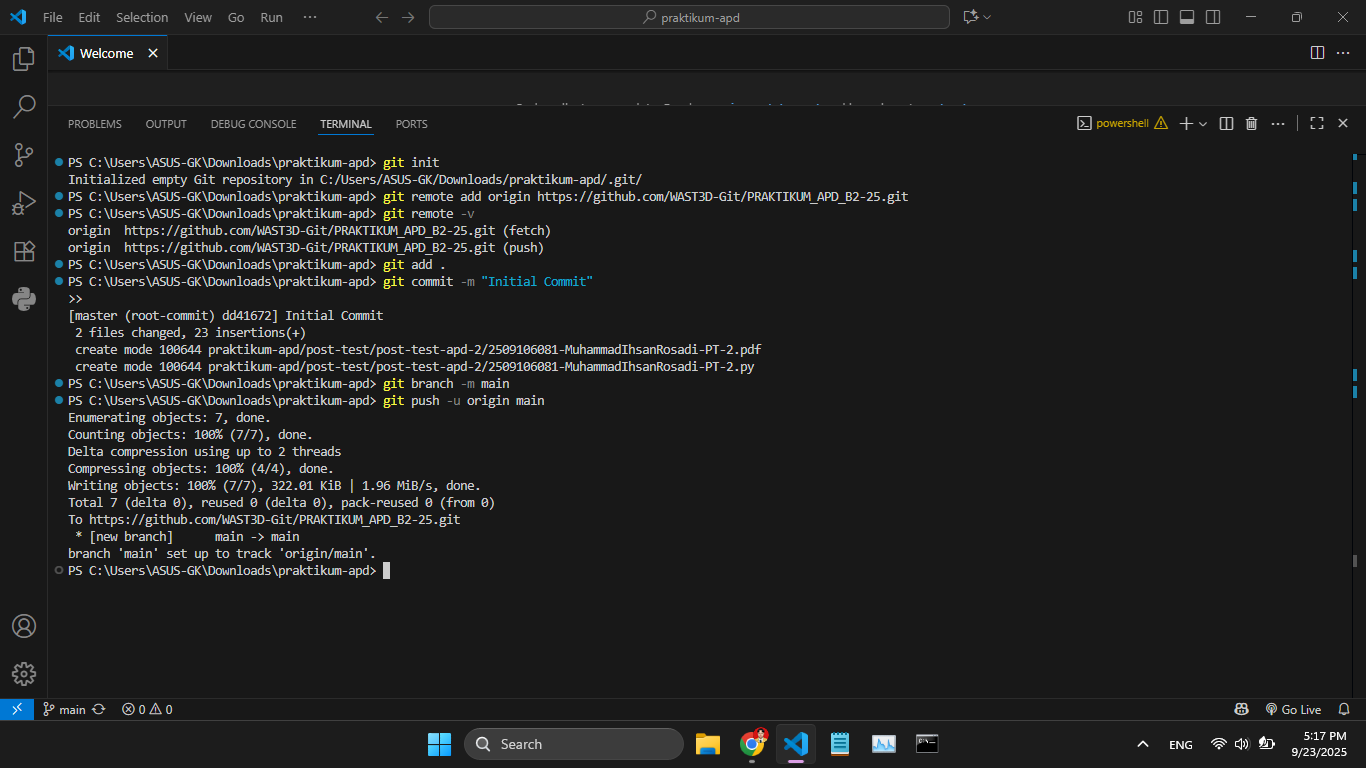


Gambar 4.17 Fitur Register (Gagal/Sudah Ada)

## 5. Langkah-langkah GIT

### **5.1 GIT Add**

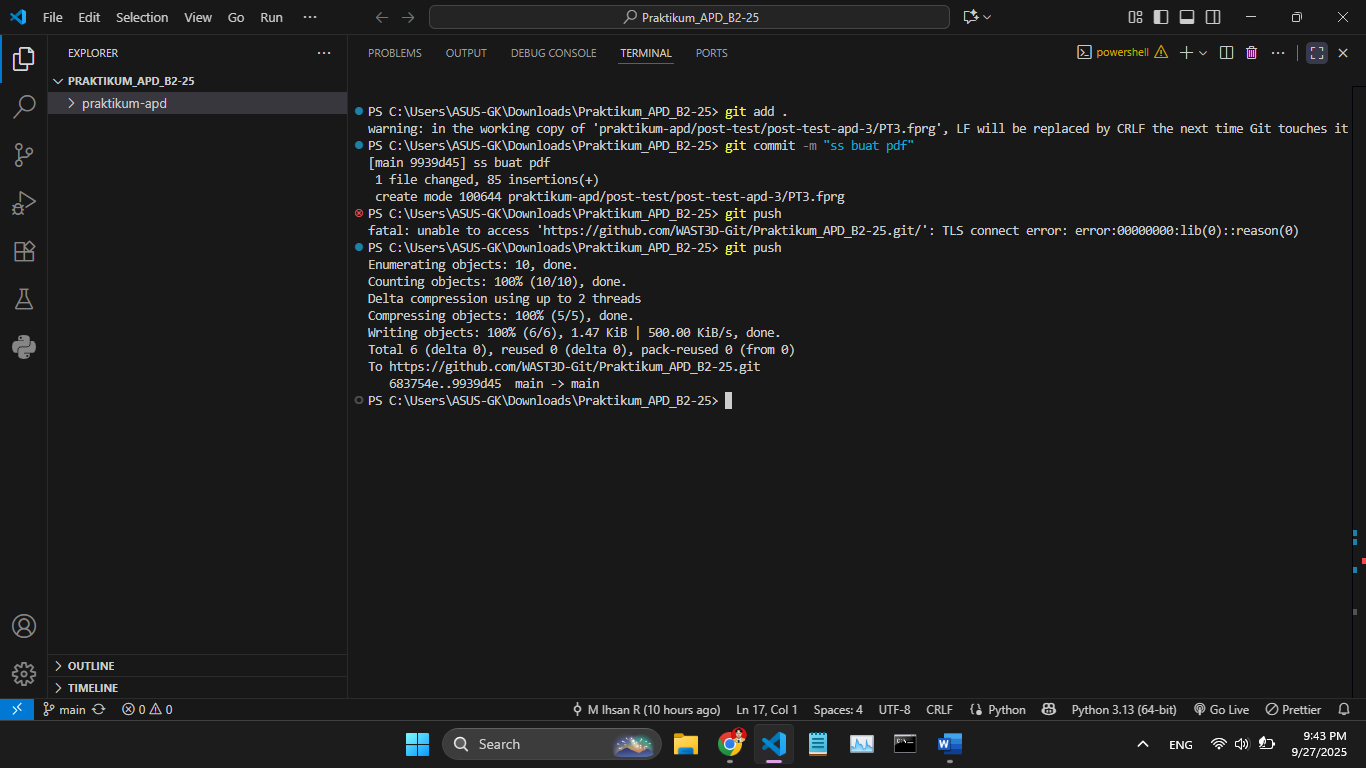
Git Add berfungsi untuk menambahkan perubahan file dari working directory ke staging area, yaitu tempat sementara sebelum perubahan benar-benar disimpan dengan git commit, dengan kata lain, perintah ini menandai file atau perubahan tertentu agar disertakan dalam commit berikutnya.



Gambar 5.1.1 Git Add

### **5.2 GIT Commit**

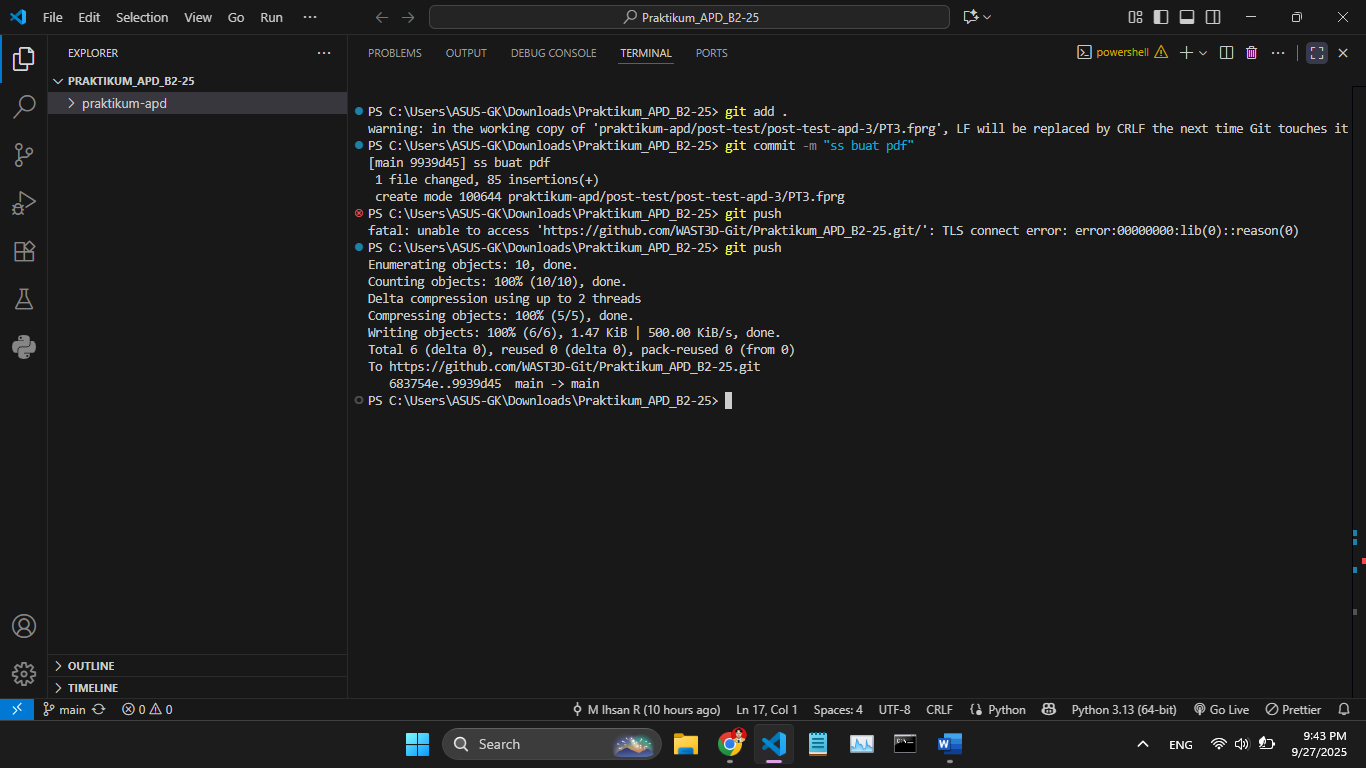
Git Commit berfungsi untuk menyimpan perubahan yang telah ditandai dengan git add ke dalam riwayat repository lokal, sehingga setiap versi proyek terdokumentasi dengan jelas melalui pesan commit yang menjelaskan perubahan tersebut.



Gambar 5.2.1 Git Commit

### **5.3 GIT Push**

Git Push adalah perintah Git yang digunakan untuk mengirim atau mengunggah perubahan dari repository lokal ke repository remote, sehingga commit yang sudah dibuat secara lokal dapat tersinkronisasi dan tersedia di server (misalnya GitHub atau GitLab) agar bisa diakses, digunakan, atau dikolaborasikan oleh orang lain.



Gambar 5.3.1 Git Push