

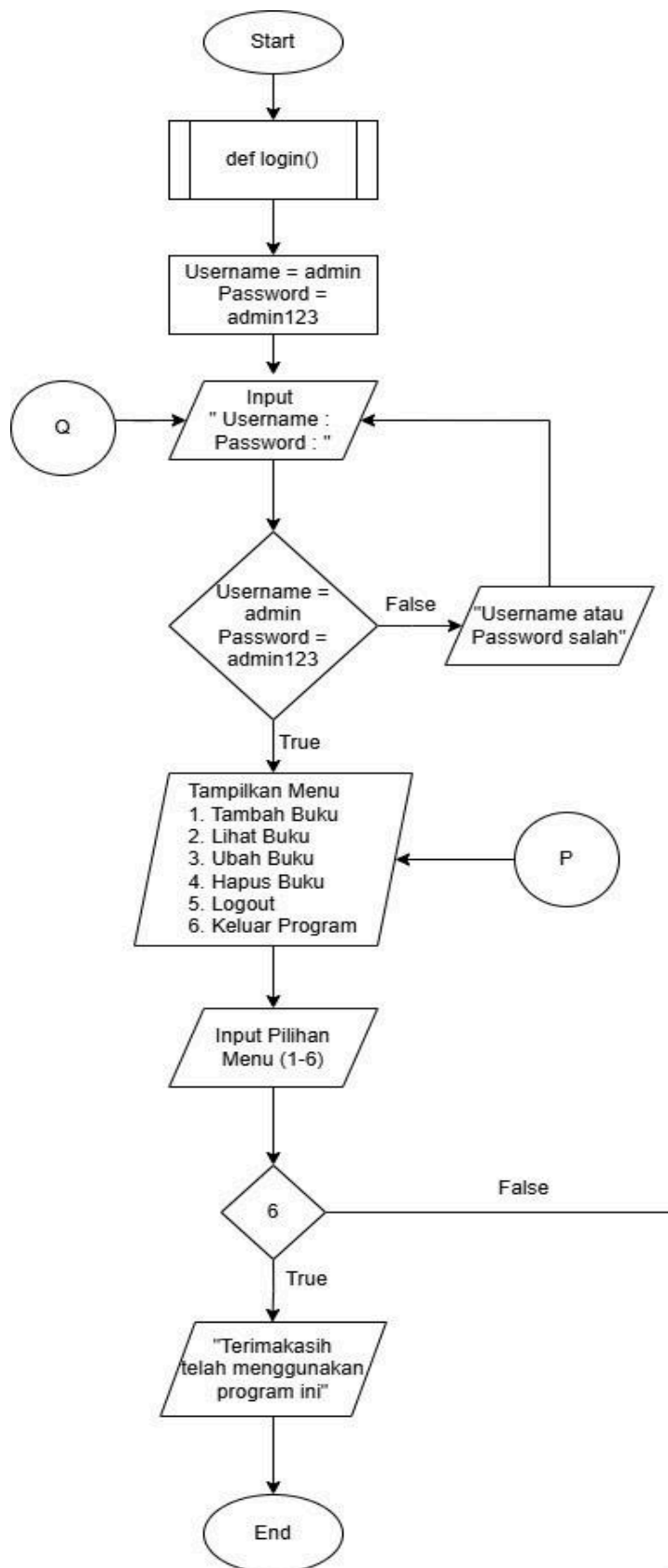
LAPORAN PRAKTIKUM
POSTTEST (7)
ALGORITMA PEMROGRAMAN DASAR

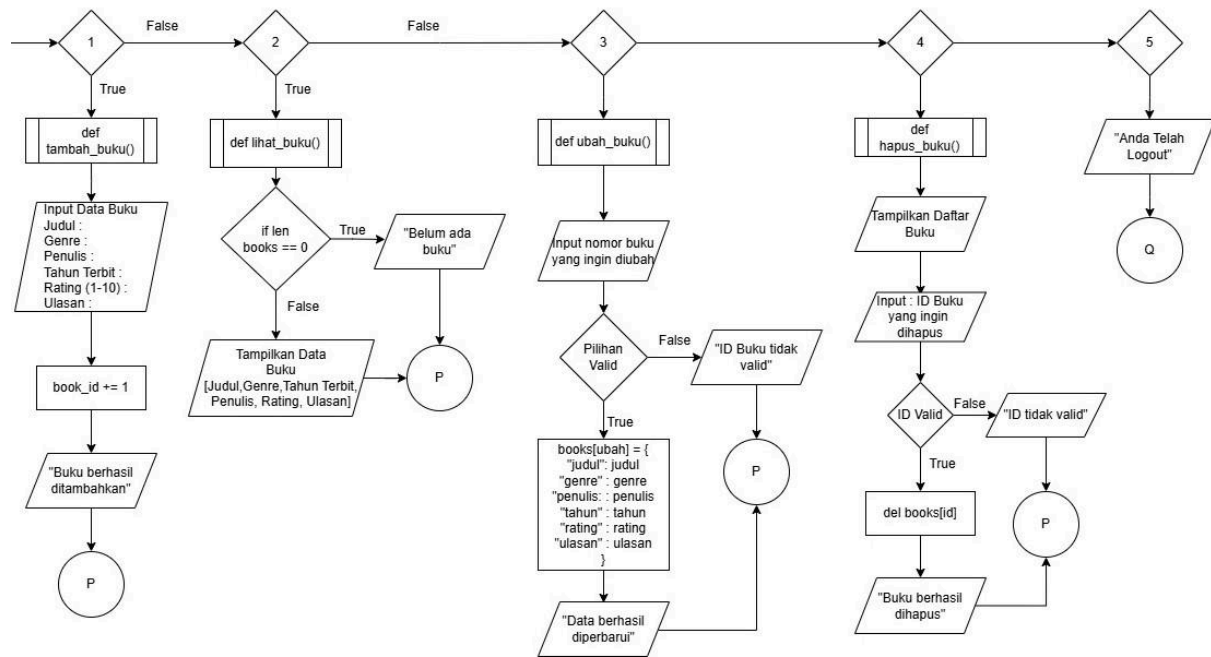


Disusun oleh:
Ahmad Afif Adiyatma (2509106047)
Kelas (B1'25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart





Penjelasan Flowchart

1. def login() = sebagai fungsi login
2. Output Tampilkan Menu = Memberikan tampilan menu dan pilihan opsi yang dapat dipilih pengguna
3. Input Pilihan Menu = Pengguna dapat memasukkan pilihan mereka dari opsi yang terdaftar (1-6)
4. Decision 1 = menentukan apakah pengguna memilih opsi 1, bila True lanjut ke bagian Create
5. def tambah_buku() = sebagai fungsi menambahkan buku
6. Input Data Buku = pengguna dapat menginput data buku mereka yang mengandung Judul, Genre, Tahun terbit, Penulis, Rating(1-10), dan Ulasan Pribadi mereka
7. Process book_id += 1= memasukkan nilai yang sudah di input oleh pengguna ke program
8. Output = "Buku Berhasil Ditambahkan"
9. On-page P = kembali ke tampilan menu
10. Decision 2 = Menentukan apabila pengguna memilih opsi 2, bila True lanjut ke bagian Read
11. def lihat_buku() = sebagai fungsi untuk melihat daftar dan data buku
12. Decision apakah books kosong = bila false output menampilkan "Belum ada buku", bila True Menampilkan Data Buku
13. On-page P = kembali ke tampilan menu
14. Decision 3 = Menentukan apabila pengguna memilih opsi 3, bila True lanjut ke bagian Update
15. def ubah_buku() = sebagai fungsi untuk mengubah databuku
16. Output Tampilkan daftar Buku = Menampilkan daftar buku di dalam program
17. Input nomor buku = pengguna memasukkan nomor buku yang datanya mau mereka Update/Ubah

18. Process = DI mana terjadi process data buku diubah
19. Output = "Data berhasil diperbarui"
20. On-page P = kembali ke tampilan menu
21. Decision 4 = Menentukan apabila pengguna memilih opsi 4, bila True lanjut ke bagian Delete
22. def hapus_buku() = sebagai fungsi untuk menghapus buku
23. Output Tampilkan daftar Buku = Menampilkan daftar buku di dalam program
24. Input nomor buku = pengguna memasukkan nomor buku yang ingin mereka hapus dari program
25. Process del books[ID] = process terjadinya penghapusan buku dari program
26. Output = "Buku berhasil dihapus"
27. On-page P = kembali ke tampilan menu
28. Decision 5 = menentukan apabila pengguna memilih opsi 5, bila True , muncul tampilan "Anda telah Logout"
29. On-page Q = kembali ke tampilan login
30. Decision 6 = menentukan apabila pengguna memilih opsi 6, bila True , muncul tampilan "Terimakasih telah menggunakan program ini"
31. End

2. Deskripsi Singkat Program

Program “Koleksi Buku Pribadi” adalah aplikasi sederhana berbasis Python yang digunakan untuk mencatat dan mengelola daftar buku atau novel yang telah dibaca oleh pengguna.

Program ini menerapkan konsep CRUD (Create, Read, Update, Delete) dengan menggunakan dictionary untuk menyimpan data buku seperti judul, genre, penulis, tahun terbit, rating, dan ulasan pribadi.

Setiap fitur utama, seperti login, tambah buku, lihat buku, ubah buku, dan hapus buku, dibuat menggunakan fungsi (def) agar program lebih terstruktur dan mudah dipahami.

Program juga menggunakan `os.system("cls || clear")` untuk membersihkan layar terminal sehingga tampilan tetap rapi di Windows, macOS, maupun Linux.

Sebagai tambahan, program ini telah dilengkapi dengan error handling sederhana tanpa try-except, menggunakan logika kondisi (if, else) untuk menangani kesalahan input dari pengguna — misalnya saat memilih menu yang tidak valid atau memasukkan nomor buku yang salah.

Fitur login dan logout juga disertakan untuk membatasi akses hanya bagi pengguna dengan username admin dan password admin123.

3. Source Code

```
import os

books = {
    1: {
        "judul": "The Lord of the Rings",
        "genre": "Fantasi",
        "penulis": "J.R.R. Tolkien",
        "tahun": "1954",
        "rating": "10",
        "ulasan": "Kisah epik petualangan dan makna persahabatan."
    },
    2: {
        "judul": "The Hobbit",
        "genre": "Petualangan",
        "penulis": "J.R.R. Tolkien",
        "tahun": "1937",
        "rating": "9",
        "ulasan": "Cerita petualangan yang ringan dan menyenangkan."
    }
}

username_admin = "admin"
password_admin = "admin123"

def clear_screen():
    os.system("cls || clear")
```

```
def clear_screen():  
    os.system("cls || clear")  
  
def tampilkan_menu():  
    print("\n=== KOLEKSI BUKU PRIBADI ===")  
    print("1. Tambah Buku")  
    print("2. Lihat Buku")  
    print("3. Ubah Buku")  
    print("4. Hapus Buku")  
    print("5. Logout")  
    print("6. Keluar")  
  
def login():  
    clear_screen()  
    print("=== LOGIN ADMIN ===")  
    user = input("Username: ")  
    pw = input("Password: ")  
    return user == username_admin and pw == password_admin
```

```

def tambah_buku():
    clear_screen()
    print("=== TAMBAH BUKU ===")
    try:
        judul = input("Judul: ")
        genre = input("Genre: ")
        penulis = input("Penulis: ")
        tahun = input("Tahun Terbit: ")
        rating = input("Rating (1-10): ")
        ulasan = input("Ulasan: ")
        id_baru = len(books) + 1
        books[id_baru] = {
            "judul": judul,
            "genre": genre,
            "penulis": penulis,
            "tahun": tahun,
            "rating": rating,
            "ulasan": ulasan
        }
        print("Buku berhasil ditambahkan!")
    except Exception as e:
        print("Terjadi kesalahan:", e)
    input("Tekan Enter untuk lanjut...")

```



```

def lihat_buku():
    clear_screen()
    print("=== DAFTAR BUKU ===")
    if not books:
        print("Belum ada buku.")
    else:
        for id, data in books.items():
            print(f"{id}. {data['judul']} ({data['genre']}) - {data['penulis']}")
            print(f"    Tahun: {data['tahun']}, Rating: {data['rating']/10}")
            print(f"    Ulasan: {data['ulasan']}")
            print("-" * 40)
        input("Tekan Enter untuk lanjut...")

def ubah_buku():
    clear_screen()
    print("=== UBAH DATA BUKU ===")
    if not books:
        print("Belum ada buku.")
    else:
        try:
            for id, data in books.items():
                print(f"{id}. {data['judul']}")
            pilih = int(input("Masukkan ID buku: "))
            if pilih in books:
                b = books[pilih]
                b["judul"] = input("Judul baru: ") or b["judul"]
                b["genre"] = input("Genre baru: ") or b["genre"]
                b["penulis"] = input("Penulis baru: ") or b["penulis"]
                b["tahun"] = input("Tahun baru: ") or b["tahun"]
                b["rating"] = input("Rating baru: ") or b["rating"]
                b["ulasan"] = input("Ulasan baru: ") or b["ulasan"]
                print("Data buku berhasil diubah!")
            else:
                print("ID tidak valid.")
        except ValueError:
            print("Input harus berupa angka!")
        input("Tekan Enter untuk lanjut...")

```

```

def hapus_buku():
    clear_screen()
    print("=== HAPUS DATA BUKU ===")
    if not books:
        print("Belum ada buku.")
    else:
        try:
            for id, data in books.items():
                print(f"{id}. {data['judul']}")
            hapus = int(input("Masukkan ID buku yang ingin dihapus: "))
            if hapus in books:
                del books[hapus]
                print("Buku berhasil dihapus!")
            else:
                print("ID tidak valid.")
        except ValueError:
            print("Input harus berupa angka!")
    input("Tekan Enter untuk lanjut...")

```

```

while True:
    if login():
        while True:
            clear_screen()
            tampilkan_menu()
            menu = input("Pilih menu: ")

            if menu == "1":
                tambah_buku()
            elif menu == "2":
                lihat_buku()
            elif menu == "3":
                ubah_buku()
            elif menu == "4":
                hapus_buku()
            elif menu == "5":
                print("Anda telah logout.")
                input("Tekan Enter untuk kembali ke login...")
                break
            elif menu == "6":
                print("Terima kasih telah menggunakan program ini!")
                exit()
            else:
                print("Pilihan tidak valid!")
                input("Tekan Enter untuk lanjut...")
        else:
            print("Username atau password salah!")
            input("Tekan Enter untuk coba lagi...")

```

4. Hasil Output

```
=== LOGIN ADMIN ===  
Masukkan username: admin  
Masukkan password: admin123  
Login berhasil!  
Tekan Enter untuk masuk ke menu...
```

```
=== KOLEKSI BUKU PRIBADI ===  
1. Tambah Buku  
2. Lihat Buku  
3. Ubah Buku  
4. Hapus Buku  
5. Logout  
6. Keluar Program  
Pilih menu (1-6):
```

```
=== TAMBAH BUKU ===  
Judul:  
Genre:  
Penulis:  
Tahun Terbit:  
Rating (1-10):  
Ulasan:  
Buku berhasil ditambahkan!  
Tekan Enter untuk lanjut...
```

```
=== DAFTAR BUKU ===  
1. The Lord of the Rings (Fantasy) - J.R.R. Tolkien  
   Tahun: 1954, Rating: 10/10  
   Ulasan: Kisah epik penuh petualangan dan makna persahabatan.  
-----  
2. The Hobbit (Fantasy) - J.R.R. Tolkien  
   Tahun: 1937, Rating: 9/10  
   Ulasan: Cerita petualangan yang ringan dan menyenangkan.  
-----  
3. () -  
   Tahun: , Rating: /10  
   Ulasan:  
-----  
Tekan Enter untuk lanjut...
```

```
=== UBAH DATA BUKU ===
```

```
1. The Lord of the Rings
```

```
2. The Hobbit
```

```
3.
```

```
Masukkan ID buku yang ingin diubah: █
```

```
=== HAPUS DATA BUKU ===
```

```
1. The Lord of the Rings
```

```
2. The Hobbit
```

```
3.
```

```
Masukkan ID buku yang ingin dihapus: 3
```

```
Buku berhasil dihapus!
```

```
Tekan Enter untuk lanjut... █
```

```
Anda telah logout.
```

```
Tekan Enter untuk kembali ke login... █
```

```
Terima kasih telah menggunakan program ini!
```

```
PS D:\Praktikum-APD\Post-Test\post-test-apd-6> █
```

5. Langkah-langkah GIT

```
PS D:\Praktikum-APD> git add .
PS D:\Praktikum-APD> git commit -m "percobaan"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS D:\Praktikum-APD> git push -u origin main
branch 'main' set up to track 'origin/main'.
Everything up-to-date
PS D:\Praktikum-APD> █
```

5.1 GIT Add

untuk memindahkan perubahan pada berkas dari direktori kerja ke staging area

5.2 GIT Commit

menyimpan snapshot perubahan yang telah dibuat ke dalam riwayat repositori git

5.3 GIT Push

untuk mengunggah commit dari repositori lokal ke repositori jarak jauh