

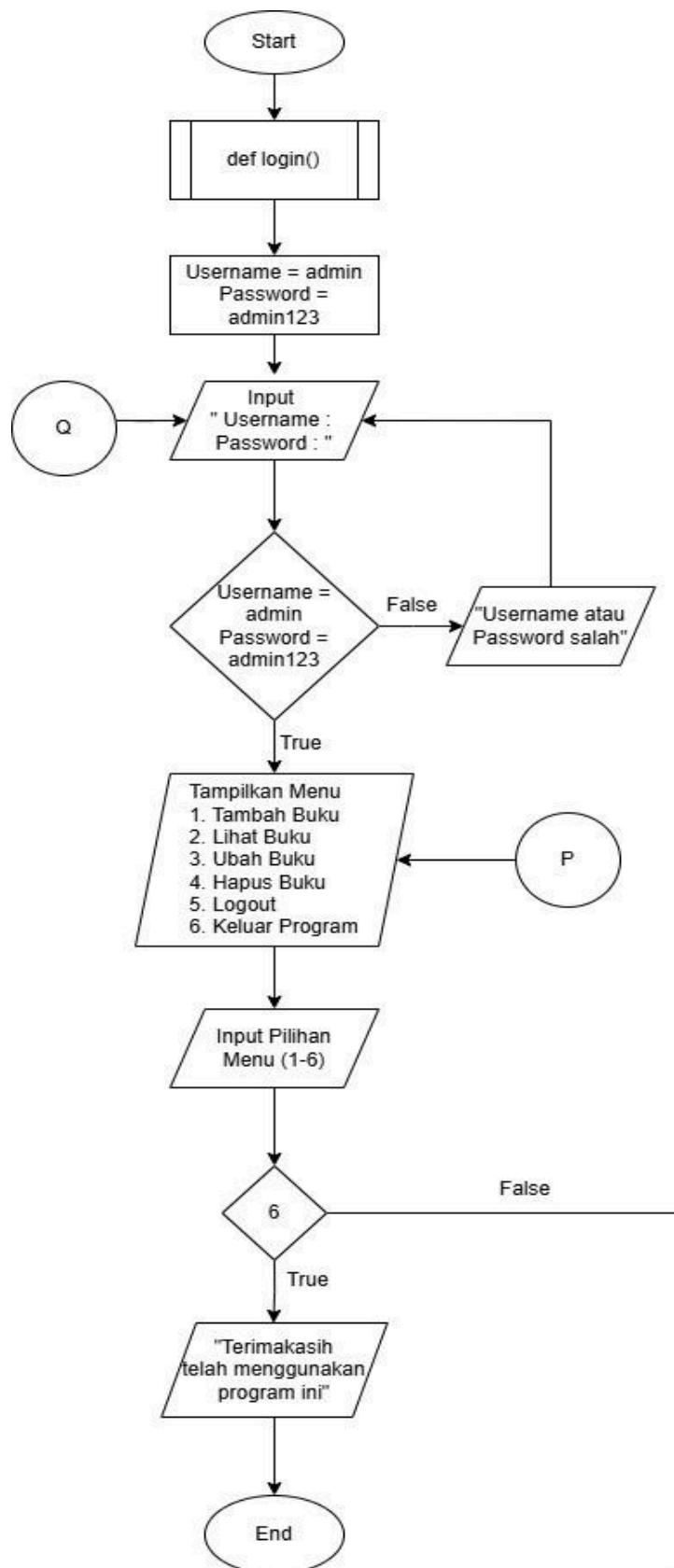
**LAPORAN PRAKTIKUM
POSTTEST (8)
ALGORITMA PEMROGRAMAN DASAR**

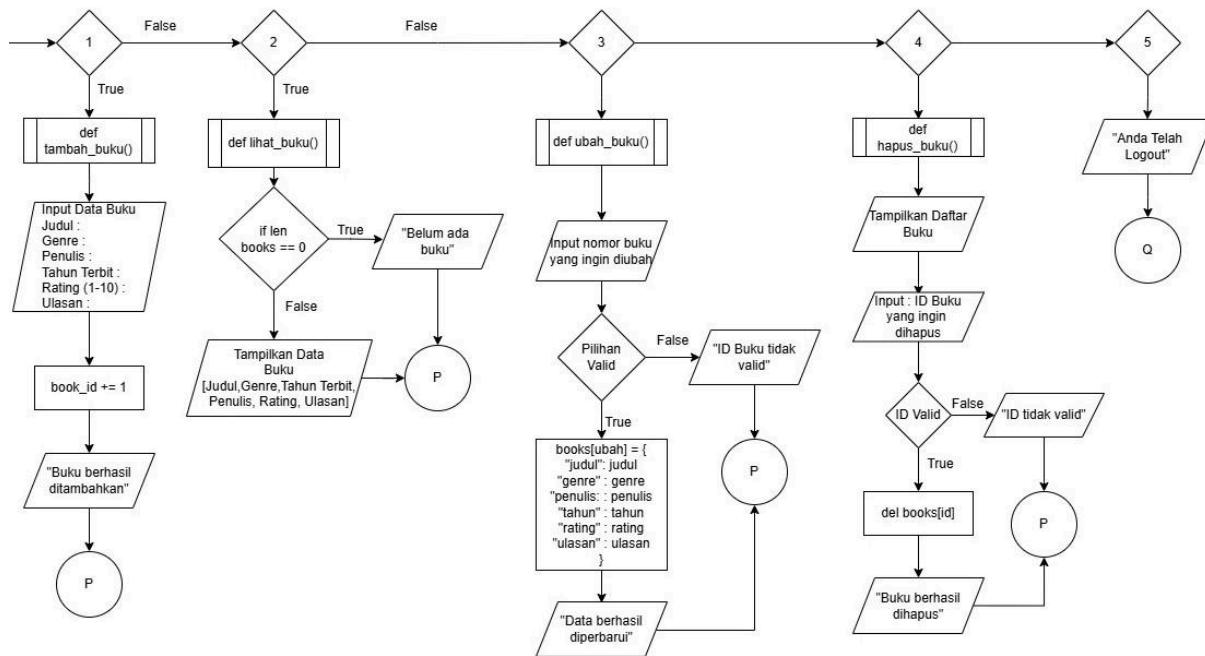


Disusun oleh:
Ahmad Afif Adiyatma (2509106047)
Kelas (B1'25)

**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 ini merupakan aplikasi sederhana berbasis Python yang berfungsi untuk mengelola koleksi buku pribadi. Aplikasi ini memungkinkan pengguna (admin) untuk menambahkan, melihat, mengubah, dan menghapus data buku menggunakan konsep CRUD (Create, Read, Update, Delete). Setiap data buku disimpan dalam struktur dictionary, sehingga mudah diakses dan diperbarui. Tampilan daftar buku ditampilkan dengan bantuan library PrettyTable dan Inquirer agar data terlihat lebih rapi dan terstruktur dalam bentuk tabel dan pengguna tidak perlu untuk input pilihan program untuk memilih. Program ini juga dilengkapi fitur login dan logout untuk keamanan, serta menggunakan clear terminal (cls || clear) agar tampilan tetap bersih. Selain itu, terdapat penerapan error handling menggunakan try dan except untuk mencegah kesalahan input.

3. Source Code

```
main.py      X  data.py      X  login.py      X  admin.py      X
main.py > ...
1 import login
2 import admin
3 import data
4 import inquirer
5
6 while True:
7     if not login.login_admin():
8         continue
9
10    while True:
11        pertanyaan = [
12            inquirer.List(
13                "menu",
14                message="Pilih menu:",
15                choices=[
16                    "Lihat Buku",
17                    "Tambah Buku",
18                    "Ubah Buku",
19                    "Hapus Buku",
20                    "Logout",
21                    "Keluar"
22                ],
23            )
24        ]
25        jawaban = inquirer.prompt(pertanyaan)
26        pilihan = jawaban["menu"]
27
28        if pilihan == "Lihat Buku":
29            if admin.lihat_buku(data.books):
30                print("\nData buku berhasil ditampilkan.")
31            else:
32                print("\nTidak ada buku untuk ditampilkan.")
33            input("\nTekan Enter untuk kembali...")
34
```

Gambar 3.1

```
    elif pilihan == "Tambah Buku":
        if admin.tambah_buku(data.books):
            print("\nBuku baru berhasil disimpan.")
        else:
            print("\nGagal menambah buku.")
        input("\nTekan Enter untuk kembali...")

    elif pilihan == "Ubah Buku":
        if admin.ubah_buku(data.books):
            print("\nData buku berhasil diperbarui.")
        else:
            print("\nTidak ada yang diubah.")
        input("\nTekan Enter untuk kembali...")

    elif pilihan == "Hapus Buku":
        admin.hapus_buku()

    elif pilihan == "Logout":
        login.logout_admin()
        break

    elif pilihan == "Keluar":
        print("\nTerima kasih sudah menggunakan program ini!")
        exit()
```

Gambar 3.2

```
main.py          data.py          login.py      admin.py
login.py > ...
1 import os
2
3 username_admin = "admin"
4 password_admin = "admin123"
5
6 def clear():
7     os.system("cls || clear")
8
9 def login_admin():
10    clear()
11    print("==> LOGIN ADMIN ==>")
12    user = input("Username: ")
13    pw = input("Password: ")
14
15    if user == username_admin and pw == password_admin:
16        print("\nLogin berhasil!")
17        input("Tekan Enter untuk lanjut...")
18        return True
19    else:
20        print("\nUsername atau password salah.")
21        input("Tekan Enter untuk coba lagi...")
22        return False
23
24 def logout_admin():
25     print("\nLogout berhasil, sampai jumpa lagi!")
26     input("Tekan Enter untuk kembali ke login...")
```

Gambar 3.3

The screenshot shows a code editor with four tabs at the top: main.py, data.py, login.py, and admin.py. The data.py tab is active, displaying the following Python code:

```
1  books = {  
2      1: {  
3          "judul": "The Lord of the Rings",  
4          "genre": "Fantasy",  
5          "penulis": "J.R.R. Tolkien",  
6          "tahun": "1954",  
7          "rating": "10",  
8          "ulasan": "Kisah epik dengan dunia yang sangat luas."  
9      },  
10     2: {  
11         "judul": "The Hobbit",  
12         "genre": "Fantasy",  
13         "penulis": "J.R.R. Tolkien",  
14         "tahun": "1937",  
15         "rating": "9",  
16         "ulasan": "Petualangan yang ringan dan menyenangkan."  
17     }  
18 }
```

Gambar 3.4

The screenshot shows a code editor with a dark theme. The file being edited is `admin.py`. The code is a Python script for managing a book database. It includes functions for displaying a main menu, clearing the screen, printing the menu, listing books, and adding books. The code uses the `PrettyTable` library to print tables of book data.

```
1 import os
2 from prettytable import PrettyTable
3 import data
4
5 menu_utama = ["Lihat Buku", "Tambah Buku", "Ubah Buku", "Hapus Buku", "Logout", "Keluar"]
6
7 def clear():
8     os.system("cls || clear")
9     (function) def tampilan_menu() -> None
10
11 def tampilan_menu():
12     clear()
13     print("== MENU UTAMA ==")
14     for i, menu in enumerate(menu_utama, 1):
15         print(f"{i}. {menu}")
16
17 def lihat_buku(books):
18     clear()
19     print("== DAFTAR BUKU ==")
20     if not books:
21         print("Belum ada buku yang tersimpan.")
22         return False
23     else:
24         tabel = PrettyTable(["ID", "Judul", "Genre", "Penulis", "Tahun", "Rating", "Ulasan"])
25         for id_buku, buku in books.items():
26             tabel.add_row([
27                 id_buku, buku["judul"], buku["genre"],
28                 buku["penulis"], buku["tahun"], buku["rating"], buku["ulasan"]
29             ])
30         print(tabel)
31         return True
```

Gambar 3.5

```

def tambah_buku(books):
    clear()
    print("== TAMBAH BUKU BARU ==")
    try:
        id_buku = input("ID Buku: ")
        if id_buku in books:
            print("ID sudah dipakai!")
            return False
        else:
            judul = input("Judul: ")
            genre = input("Genre: ")
            penulis = input("Penulis: ")
            tahun = input("Tahun Terbit: ")
            rating = input("Rating (1-10): ")
            ulasan = input("Ulasan: ")
            books[id_buku] = {
                "judul": judul, "genre": genre, "penulis": penulis,
                "tahun": tahun, "rating": rating, "ulasan": ulasan
            }
            print("\nBuku berhasil ditambahkan!")
            return True
    except:
        print("\nTerjadi kesalahan input.")
        return False
    input("Tekan Enter untuk kembali...")

def ubah_buku(books):
    clear()
    print("== UBAH DATA BUKU ==")
    if not books:
        print("Belum ada buku untuk diubah.")
        input("Tekan Enter untuk kembali...")
        return False

```

Gambar 3.6

```

for id_buku, buku in books.items():
    print(f"{id_buku}. {buku['judul']}")
pilih = input("\nMasukkan ID buku yang ingin diubah: ")

if pilih in (parameter) books: Any
    buku = books[pilih]
    print("Kosongkan jika tidak ingin diubah.")
    judul = input(f"Judul baru [{buku['judul']}]: ") or buku["judul"]
    genre = input(f"Genre baru [{buku['genre']}]: ") or buku["genre"]
    penulis = input(f"Penulis baru [{buku['penulis']}]: ") or buku["penulis"]
    tahun = input(f"Tahun baru [{buku['tahun']}]: ") or buku["tahun"]
    rating = input(f"Rating baru [{buku['rating']}]: ") or buku["rating"]
    ulasan = input(f"Ulasan baru [{buku['ulasan']}]: ") or buku["ulasan"]
    books[pilih] = {
        "judul": judul, "genre": genre, "penulis": penulis,
        "tahun": tahun, "rating": rating, "ulasan": ulasan
    }
    print("\nData buku berhasil diubah!")
    return True
else:
    print("ID tidak ditemukan.")
    return False

```

Gambar 3.7

```

def hapus_buku():
    clear()
    print("== HAPUS BUKU ==")
    if not data.books:
        print("Belum ada buku yang bisa dihapus.")
    else:
        for id_buku, buku in data.books.items():
            print(f"{id_buku}. {buku['judul']}")
        hapus = input("\nMasukkan ID buku yang ingin dihapus: ")
        if hapus in data.books:
            konfirmasi = input(f"Yakin hapus '{data.books[hapus]['judul']}'? (y/n): ").lower()
            if konfirmasi (variable) books: dict[int, dict[str, str]]
                del data.books[hapus]
                print("Buku berhasil dihapus!")
            else:
                print("Penghapusan dibatalkan.")
        else:
            print("ID tidak ditemukan.")
    input("\nTekan Enter untuk kembali...")

```

Gambar 3.8

4. Hasil Output

```
==== LOGIN ADMIN ====
Username: admin
Password: admin123

Login berhasil!
Tekan Enter untuk lanjut...
[?] Pilih menu::
    > Lihat Buku
        Tambah Buku
        Ubah Buku
        Hapus Buku
        Logout
        Keluar
```

Gambar 4.1

```
==== DAFTAR BUKU ====
+-----+
| ID | Judul           | Genre | Penulis      | Tahun | Rating | Ulasan          |
+-----+
| 1  | The Lord of the Rings | Fantasy | J.R.R. Tolkien | 1954 | 10     | Kisah epik dengan dunia yang sangat luas. |
| 2  | The Hobbit        | Fantasy | J.R.R. Tolkien | 1937 | 9      | Petualangan yang ringan dan menyenangkan. |
+-----+

Data buku berhasil ditampilkan.

Tekan Enter untuk kembali...■
```

Gambar 4.2

```
==== TAMBAH BUKU BARU ====
ID Buku: 3
Judul:
Genre:
Penulis:
Tahun Terbit:
Rating (1-10):
Ulasan:

Buku berhasil ditambahkan!

Buku baru berhasil disimpan.

Tekan Enter untuk kembali...■
```

Gambar 4.3

```

==== UBAH DATA BUKU ===
1. The Lord of the Rings
2. The Hobbit
3.

Masukkan ID buku yang ingin diubah: 3
Kosongkan jika tidak ingin diubah.
Judul baru []: The Silmarillion
Genre baru []: Fantasi
Penulis baru []: J.R.R Tolkien
Tahun baru []: 1977
Rating baru []: 8
Ulasan baru []: Koleksi cerita di dunia The Hobbit dan LOTR

Data buku berhasil diubah!

Data buku berhasil diperbarui.

Tekan Enter untuk kembali...

```

Gambar 4.4

```

==== DAFTAR BUKU ===
+-----+-----+-----+-----+-----+-----+
| ID | Judul | Genre | Penulis | Tahun | Rating | Ulasan |
+-----+-----+-----+-----+-----+-----+
| 1 | The Lord of the Rings | Fantasy | J.R.R. Tolkien | 1954 | 10 | Kisah epik dengan dunia yang sangat luas. |
| 2 | The Hobbit | Fantasy | J.R.R. Tolkien | 1937 | 9 | Petualangan yang ringan dan menyenangkan. |
| 3 | The Silmarillion | Fantasi | J.R.R Tolkien | 1977 | 8 | Koleksi cerita di dunia The Hobbit dan LOTR |
+-----+-----+-----+-----+-----+-----+
Data buku berhasil ditampilkan.

Tekan Enter untuk kembali...

```

Gambar 4.5

```
==== HAPUS BUKU ====
1. The Lord of the Rings
2. The Hobbit
3. The Silmarillion

Masukkan ID buku yang ingin dihapus: 3
Yakin hapus 'The Silmarillion'? (y/n): n
Penghapusan dibatalkan.

Tekan Enter untuk kembali... █
```

Gambar 4.6

```
Tekan Enter untuk kembali...
[?] Pilih menu::
    Lihat Buku
    Tambah Buku
    Ubah Buku
    Hapus Buku
    Logout
> Keluar

Terima kasih sudah menggunakan program ini!
PS D:\Praktikum-APD\Post-Test\2509106047-Ahmadafifadiyatma-PT-8> █
```

Gambar 4.7

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