

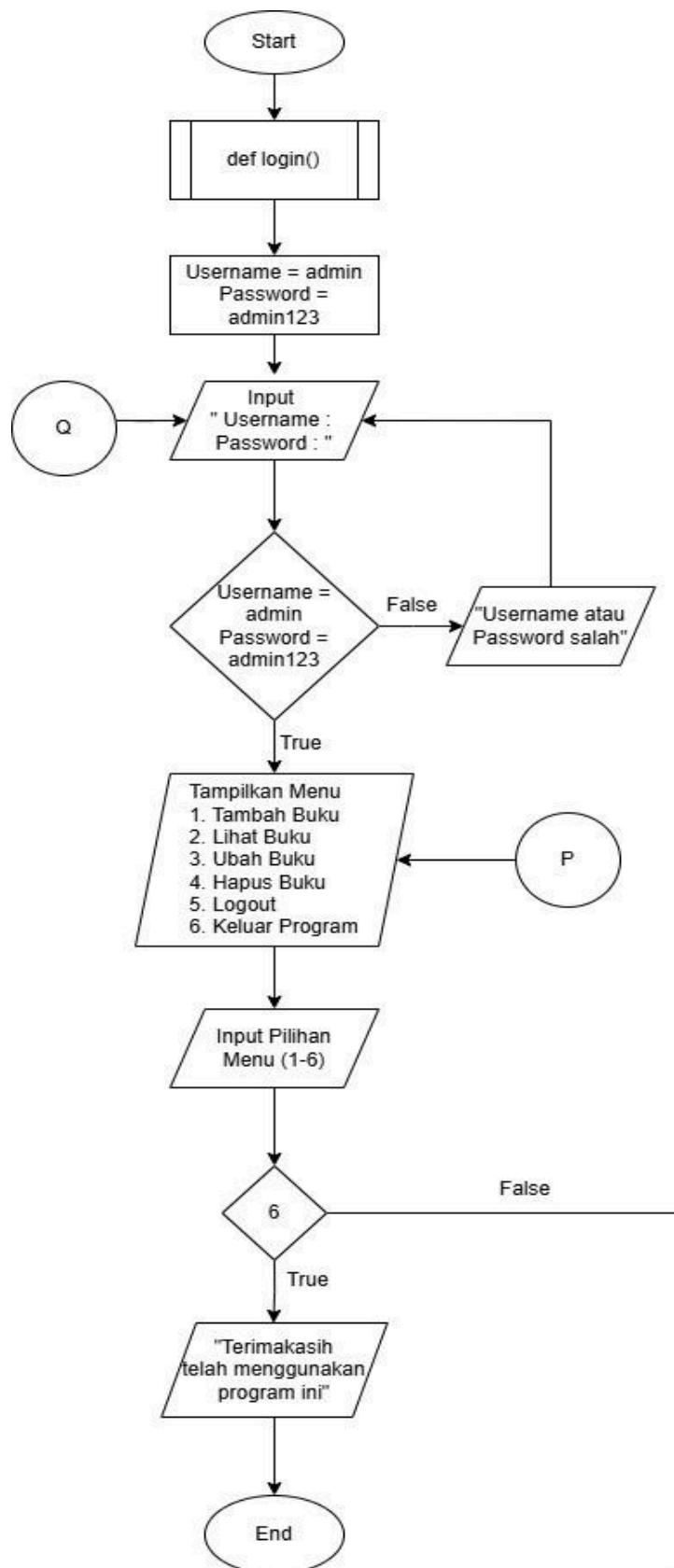
**LAPORAN PRAKTIKUM
POSTTEST (8)
ALGORITMA PEMROGRAMAN DASAR**

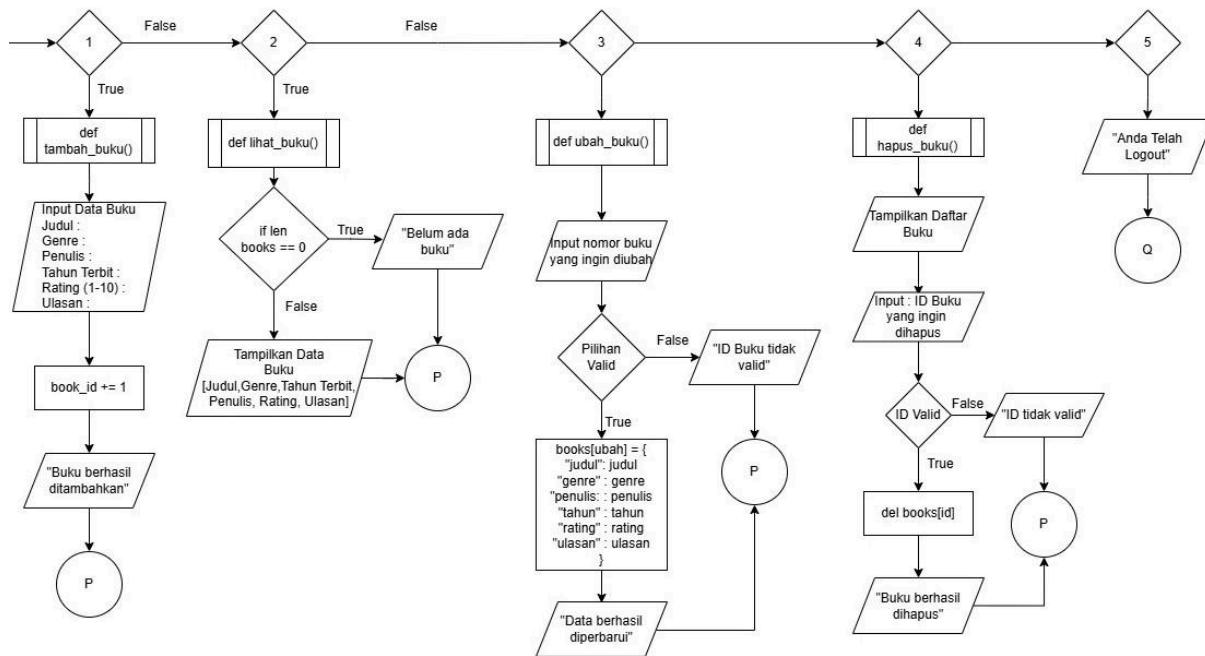


Disusun oleh:
Ahmad Afif Adiyatma (2509106047)
Kelas (B1'25)

**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 agar data terlihat lebih rapi dan terstruktur dalam bentuk tabel. 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

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

```
 1 import login
 2 import admin
 3
 4 while True:
 5     if not login.login_admin():
 6         continue
 7
 8     while True:
 9         login.clear()
10         print("== MENU UTAMA ==")
11         print("1. Lihat Buku")
12         print("2. Tambah Buku")
13         print("3. Ubah Buku")
14         print("4. Hapus Buku")
15         print("5. Logout")
16         print("6. Keluar")
17
18         pilih = input("Pilih menu (1-6): ")
19
20         if pilih == "1":
21             admin.lihat_buku()
22         elif pilih == "2":
23             admin.tambah_buku()
24         elif pilih == "3":
25             admin.ubah_buku()
26         elif pilih == "4":
27             admin.hapus_buku()
28         elif pilih == "5":
29             print("\nLogout berhasil.")
30             break
31         elif pilih == "6":
32             print("\nTerima kasih telah menggunakan program ini.")
33             exit()
34         else:
35             print("Pilihan tidak valid!")
36             input("Tekan Enter untuk lanjut...")
```

Gambar 3.1

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

```
login.py > login_admin
1 import os
2
3 def clear():
4     os.system("cls || clear")
5
6 def login_admin():
7     clear()
8     print("==> LOGIN ADMIN ==>")
9     user = input("Masukkan username: ")
10    pw = input("Masukkan password: ")
11
12    if user == "admin" and pw == "admin123":
13        print("\nLogin berhasil! Selamat datang.")
14        input("Tekan Enter untuk lanjut...")
15        return True
16    else:
17        print("\nUsername atau password salah.")
18        input("Tekan Enter untuk coba lagi...")
19        return False
```

Gambar 3.2

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 script:

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

Gambar 3.3

```
main.py    data.py    login.py    admin.py x
❸ admin.py > ⌂ hapus_buku
1  import os
2  from prettytable import PrettyTable
3  import data
4
5  def clear():
6      os.system("cls || clear")
7
8  def lihat_buku():
9      clear()
10     print("== DAFATAR BUKU ==")
11     if not data.books:
12         print("Belum ada buku yang tersimpan.")
13     else:
14         tabel = PrettyTable(["ID", "Judul", "Genre", "Penulis", "Tahun", "Rating", "Ulasan"])
15         for id_buku, buku in data.books.items():
16             tabel.add_row([id_buku, buku["judul"], buku["genre"], buku["penulis"], buku["tahun"], buku["rating"], buku["ulasan"]])
17     print(tabel)
18     input("\nTekan Enter untuk kembali ke menu...")
```

Gambar 3.4

```
def tambah_buku():
    clear()
    print("== TAMBAH BUKU ==")
    id_buku = input("Masukkan ID buku: ")
    if id_buku in data.books:
        print("ID sudah digunakan!")
    else:
        judul = input("Judul: ")
        genre = input("Genre: ")
        penulis = input("Penulis: ")
        tahun = input("Tahun Terbit: ")
        rating = input("Rating (1-10): ")
        ulasan = input("Ulasan: ")
        data.books[id_buku] = {
            "judul": judul,
            "genre": genre,
            "penulis": penulis,
            "tahun": tahun,
            "rating": rating,
            "ulasan": ulasan
        }
        print("Buku berhasil ditambahkan!")
        input("\nTekan Enter untuk kembali...")
```

Gambar 3.5

```

def ubah_buku():
    clear()
    print("== UBAH BUKU ==")
    if not data.books:
        print("Belum ada buku.")
    else:
        lihat_buku()
        pilih = input("Masukkan ID buku yang ingin diubah: ")
        if pilih in data.books:
            buku = data.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"]
            data.books[pilih] = {
                "judul": judul,
                "genre": genre,
                "penulis": penulis,
                "tahun": tahun,
                "rating": rating,
                "ulasan": ulasan
            }
            print("Data berhasil diperbarui!")
        else:
            print("ID buku tidak ditemukan.")
    input("\nTekan Enter untuk kembali...")

```

Gambar 3.6

```

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

```

Gambar 3.7

4. Hasil Output

```
==== LOGIN ADMIN ====
Masukkan username: admin
Masukkan password: admin123

Login berhasil! Selamat datang kembali.
Tekan Enter untuk melanjutkan...█
```

Gambar 4.1

```
==== MENU UTAMA ====
1. Lihat Koleksi Buku
2. Tambah Buku Baru
3. Ubah Data Buku
4. Hapus Buku
5. Logout
6. Keluar

Pilih menu (1-6): █
```

Gambar 4.2

```
==== DAFTAR KOLEKSI 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. |
+-----+
Tekan Enter untuk kembali...█
```

Gambar 4.3

```
==== TAMBAH BUKU BARU ===
Masukkan ID Buku: 3
Judul: The Silmarillion
Genre: Fantasi
Penulis: J.R.R Tolkien
Tahun Terbit: 1977
Rating (1-10): 7
Ulasan singkat: Di dunia yang sama dari LOTR dan The Hobbit tapi merupakan kumpulan cerita

Buku baru berhasil ditambahkan!

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

Gambar 4.4

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 7 Di dunia yang sama dari LOTR dan The Hobbit tapi merupakan kumpulan cerita						

Tekan Enter untuk kembali...

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 [7]: 8
 Ulasan baru [Di dunia yang sama dari LOTR dan The Hobbit tapi merupakan kumpulan cerita]:

Data buku berhasil diperbarui!

Tekan Enter untuk melanjutkan...■

Gambar 4.5

== DAFTAR KOLEKSI 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 Di dunia yang sama dari LOTR dan The Hobbit tapi merupakan kumpulan cerita						

Tekan Enter untuk kembali...

Masukkan ID buku yang ingin dihapus:
 Yakin ingin hapus buku ''? (y/n): y

Buku berhasil dihapus.

Tekan Enter untuk kembali...■

Gambar 4.6

== MENU UTAMA ==

1. Lihat Koleksi Buku
2. Tambah Buku Baru
3. Ubah Data Buku
4. Hapus Buku
5. Logout
6. Keluar

Pilih menu (1-6): 6

Terima kasih telah menggunakan program ini.

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