

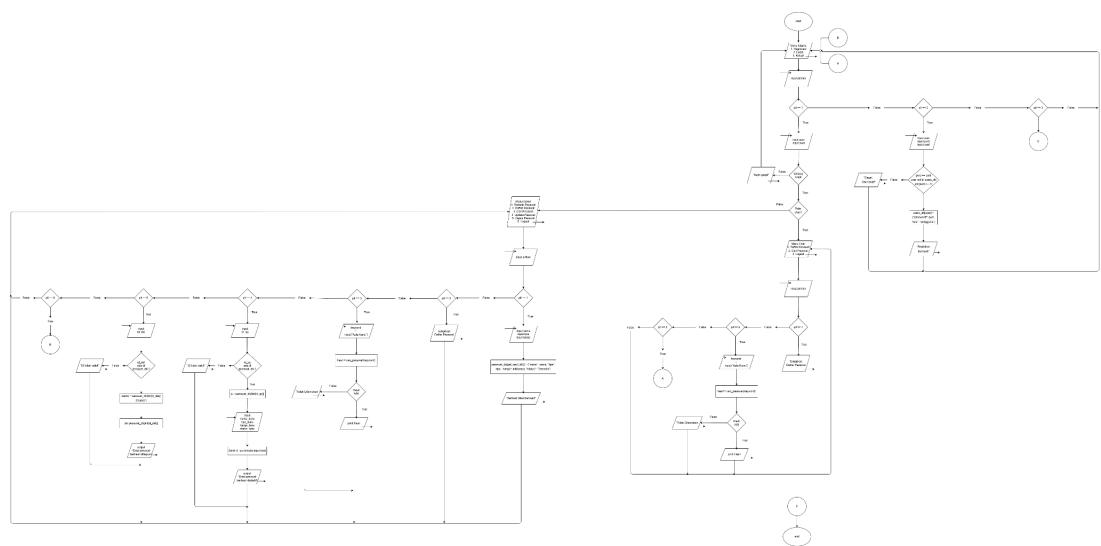
**LAPORAN PRAKTIKUM**  
**POSTTEST 8**  
**ALGORITMA PEMROGRAMAN DASAR**



**Disusun oleh:**  
**Muhammad Fahrizal Dani (2509106071)**  
**Kelas (B2 '25)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

## 1. Flowchart



Gambar 1.1 Flowchart program  
manajemen penjualan pesawat

## **Penjelasan Flowchart**

### **1. Start & Inisialisasi**

- Program dimulai dari Start.
- Dua database utama dibuat:
  - users\_db → berisi data username, password, dan role pengguna (admin atau pengguna).
  - pesawat\_db → menyimpan daftar pesawat berupa ID, nama, tipe, harga, dan status
- Variabel kontrol seperti user\_login, user\_role, dan menu\_login diset dalam kondisi awal (belum login).

### **2. Menu Login (Loop Utama)**

- Selama menu\_login == True, program akan terus menampilkan pilihan menu:
  1. Login
  2. Register
  3. Keluar
- Pengguna akan diminta memilih salah satu opsi.
- Struktur ini membentuk loop utama agar setelah selesai login/logout, sistem bisa kembali ke menu awal tanpa menutup program.

### **3. Proses Login**

Jika pengguna memilih Login, alurnya adalah sebagai berikut:

1. Pengguna menginput username dan password.
2. Sistem mencocokkan data yang dimasukkan dengan isi dari users\_db.
3. Jika data tidak ditemukan, muncul pesan "*Login gagal*" dan kembali ke menu login.
4. Jika cocok, pengguna berhasil login dan role-nya diperiksa.

## **A. Jika Role = "Admin"**

Masuk ke Menu Admin, berisi beberapa opsi utama:

1. Tambah Pesawat Baru
  - Input data pesawat baru (nama, tipe, harga, status).
  - Data tersebut disimpan ke pesawat\_db dengan ID unik.
2. Lihat Semua Pesawat
  - Menampilkan seluruh daftar pesawat yang tersimpan di database.
3. Update Data Pesawat
  - Admin memasukkan ID pesawat yang ingin diubah.
  - Jika ditemukan, admin dapat memperbarui nama, tipe, harga, atau status.
4. Hapus Data Pesawat
  - Admin memasukkan ID pesawat yang ingin dihapus.
  - Jika ID valid, data tersebut dihapus dari pesawat\_db.
5. Cari Pesawat / Laporan (opsional sesuai flowchart)
  - Menampilkan data berdasarkan kriteria tertentu (misalnya nama atau harga).
6. Logout
  - Kembali ke menu login utama.

Setiap opsi di dalam menu admin terhubung dengan proses CRUD (Create, Read, Update, Delete) terhadap pesawat\_db.

## **B. Jika Role = "Pengguna" (User)**

Masuk ke Menu User, dengan opsi yang lebih terbatas:

1. Lihat Semua Pesawat
  - Menampilkan daftar pesawat yang tersedia.
2. Logout
  - Kembali ke menu login utama.

Pengguna biasa tidak dapat menambah, mengubah, atau menghapus data pesawat.

## **4. Proses Register**

Jika pengguna memilih Register, langkahnya adalah:

1. Input username dan password baru.
2. Sistem menambahkan data tersebut ke users\_db dengan role default "pengguna".
3. Setelah registrasi berhasil, sistem akan menampilkan pesan konfirmasi.
4. Program kembali ke menu login utama.

## 5. Keluar dari Program

Jika pengguna memilih Keluar (3) pada menu login:

- Nilai menu\_login diubah menjadi False.
- Program menampilkan pesan “Terima kasih telah menggunakan program ini”.
- Proses berakhir di simbol End.

## 2. Deskripsi Singkat Program

Program Sistem Manajemen Data Pesawat adalah aplikasi berbasis console untuk mengelola data pesawat dengan fitur login, register, dan manajemen data sesuai peran pengguna. Admin dapat menambah, melihat, mencari, mengubah, dan menghapus data pesawat, sedangkan pengguna biasa hanya dapat melihat dan mencari. Program menggunakan fungsi, prosedur, serta percabangan dan perulangan untuk menjalankan menu secara interaktif, dengan validasi input dan penanganan error agar sistem berjalan stabil dan terstruktur serta code terpisah dan modular untuk meningkatkan readability.

### 3. Source Code

#### 3.1. Variabel global dictionary berisi username, password, data pesawat, dan Varibel global kondisi login di data\_data.py

```
users_db = {  
    "admin": {"password": "admin123", "role": "admin"},  
    "user1": {"password": "user123", "role": "pengguna"}  
}  
  
pesawat_db = {  
    1: {"nama": "Boeing 737 MAX 8", "tipe": "Komersial", "harga":  
        1250000000, "status": "Tersedia"},  
    2: {"nama": "Airbus A320neo", "tipe": "Komersial", "harga": 1180000000,  
        "status": "Tersedia"},  
    3: {"nama": "Cessna Citation X", "tipe": "Jet Pribadi", "harga":  
        230000000, "status": "Tersedia"},  
    4: {"nama": "Gulfstream G700", "tipe": "Jet Pribadi", "harga":  
        760000000, "status": "Tersedia"},  
    5: {"nama": "ATR 72-600", "tipe": "Regional", "harga": 250000000,  
        "status": "Tersedia"},  
    6: {"nama": "LM F-35 Lightning II", "tipe": "Militer", "harga":  
        1850000000, "status": "Tersedia"},  
    7: {"nama": "Boeing 777-300ER", "tipe": "Komersial", "harga":  
        1800000000, "status": "Tersedia"},  
    8: {"nama": "Airbus A350-1000", "tipe": "Komersial", "harga":  
        1750000000, "status": "Tersedia"},  
    9: {"nama": "Dassault Falcon 8X", "tipe": "Jet Pribadi", "harga":  
        580000000, "status": "Tersedia"},  
    10: {"nama": "Pilatus PC-24", "tipe": "Jet Pribadi", "harga": 110000000,  
        "status": "Tersedia"},  
  
    user_session = {"username": None, "role": None}
```

#### 3.2. Fungsi Parameter Validasi Login di autentikasi.py

```
from data_data import users_db, user_session  
  
def validasi_login(username, password):  
    try:  
        valid = username in users_db and users_db[username]["password"] ==  
            password  
        data = users_db[username] if valid else None  
        return valid, data  
    except:
```

### **3.3. Fungsi Parameter Register User di autentikasi.py**

```
def register_user(username, password, confirm_password):
    try:
        if not username or not password:
            return False, "Username dan password tidak boleh kosong!"

        if password != confirm_password:
            return False, "Password dan konfirmasi tidak cocok!"

        if username in users_db:
            return False, "Username sudah terdaftar!"

        if len(password) < 5:
            return False, "Password minimal 5 karakter!"

        users_db[username] = {"password": password, "role": "pengguna"}
        return True, "Registrasi berhasil!"
    except Exception as e:
        return False, f"Error: {e}"
```

### **3.4. Fungsi Parameter mengubah kondisi login di autentikasi.py**

```
def login_user(username, role):
    user_session["username"] = username
    user_session["role"] = role
```

### **3.5. Fungsi tanpa Parameter logout di autentikasi.py**

```
def logout_user():
    user_session["username"] = None
    user_session["role"] = None
```

### 3.6. Fungsi Parameter Cari Pesawat di pesawat.py

```
from data_data import pesawat_db

def cari_pesawat(keyword):
    try:
        hasil = {id: data for id, data in pesawat_db.items()
                 if keyword.lower() in data["nama"].lower() or
keyword.lower() in data["tipe"].lower()}
        return hasil
    except:
        return {}
```

### 3.7. Fungsi Tanpa Parameter Menghitung Total Pesawat di pesawat.py

```
def hitung_total_pesawat():
    try:
        total = len(pesawat_db)
        nilai = sum(p["harga"] for p in pesawat_db.values())
        return total, nilai
    except:
        return 0, 0
```

### 3.8. Fungsi Tanpa Parameter Mendapatkan Id Pesawat Selanjutnya di pesawat.py

```
def get_next_id():
    try:
        return max(pesawat_db.keys()) + 1 if pesawat_db else 1
    except:
        return 1
```

### 3.9. Fungsi Parameter Menambah Pesawat di pesawat.py

```
def tambah_pesawat(nama, tipe, harga):
    try:
        if not nama or not tipe:
            return False, "Nama dan tipe tidak boleh kosong!"

        if not harga.isdigit() or int(harga) <= 0:
            return False, "Harga harus berupa angka positif!"

        pesawat_db[get_next_id()] = {
            "nama": nama,
            "tipe": tipe,
            "harga": int(harga),
            "status": "Tersedia"
        }
        return True, "Pesawat berhasil ditambahkan!"
    except Exception as e:
```

### 3.10. Fungsi Parameter Update Pesawat di pesawat.py

```
def update_pesawat(id_pesawat, nama=None, tipe=None, harga=None,
status=None):
    try:
        if id_pesawat not in pesawat_db:
            return False, "ID pesawat tidak ditemukan!"

        p = pesawat_db[id_pesawat]

        if nama: p["nama"] = nama
        if tipe: p["tipe"] = tipe
        if harga and harga.isdigit(): p["harga"] = int(harga)
        if status: p["status"] = status

        return True, "Pesawat berhasil diupdate!"
    except Exception as e:
        return False, f"Error: {e}"
```

### **3.11. Fungsi Parameter Hapus Pesawat di pesawat.py**

```
def hapus_pesawat(id_pesawat):
    try:
        if id_pesawat not in pesawat_db:
            return False, "ID pesawat tidak ditemukan!"

        nama = pesawat_db[id_pesawat]["nama"]
        del pesawat_db[id_pesawat]
        return True, f"Pesawat '{nama}' berhasil dihapus!"
    except Exception as e:
        return False, f"Error: {e}"
```

### **3.12. Fungsi Parameter Memilih Pesawat di pesawat.py**

```
def get_pesawat(id_pesawat):
    return pesawat_db.get(id_pesawat, None)
```

### **3.13. Procedure untuk clear terminal di tampilan\_ui.py**

```
import os
from prettytable import PrettyTable
from data_data import pesawat_db
from pesawat import hitung_total_pesawat

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

### **3.14. Procedure Tampilkan Header di tampilan\_ui.py**

```
def tampilkan_header(judul):
    print("=" * 80)
    print(judul.center(80))
    print("=" * 80)
```

### 3.15. Procedure Tampilkan Daftar Pesawat di tampilan\_ui.py

```
def tampilan_daftar_pesawat():
    try:
        if not pesawat_db:
            print("Tidak ada data pesawat.")
            return

        table = PrettyTable()
        table.field_names = ["ID", "Nama Pesawat", "Tipe", "Harga (Rp)",
        "Status"]

        table.align["ID"] = "c"
        table.align["Nama Pesawat"] = "l"
        table.align["Tipe"] = "l"
        table.align["Harga (Rp)"] = "r"
        table.align["Status"] = "c"

        for id_p, p in pesawat_db.items():
            table.add_row([
                id_p,
                p['nama'],
                p['tipe'],
                f"{p['harga']:,}",
                p['status']
            ])

        print(table)

        total, nilai = hitung_total_pesawat()
        print(f"\n{'='*80}")
        print(f"Total Pesawat: {total} unit | Total Nilai: Rp {nilai:,}")
        print(f"{'='*80}")

    except Exception as e:
        print(f"Error: {e}")
```

### 3.16. Fungsi parameter Tampilkan Hasil Pencarian Pesawat di tampilan\_ui.py

```
def tampilan_hasil_pencarian(hasil):
    if hasil:
        table = PrettyTable()
        table.field_names = ["ID", "Nama Pesawat", "Tipe", "Harga (Rp)"]

        table.align["ID"] = "c"
        table.align["Nama Pesawat"] = "l"
        table.align["Tipe"] = "l"
        table.align["Harga (Rp)"] = "r"

        for id_p, p in hasil.items():
            table.add_row([
                id_p,
                p['nama'],
                p['tipe'],
                f"{p['harga']}:,}"
            ])

        print(table)
        print(f"\nDitemukan {len(hasil)} pesawat")
    else:
        print("X Tidak ditemukan.")
```

### 3.17. Fungsi parameter Tampilkan Hasil Pencarian User Pesawat di tampilan\_ui.py

```
def tampilan_hasil_pencarian_user(hasil):
    if hasil:
        table = PrettyTable()
        table.field_names = ["ID", "Nama Pesawat", "Tipe"]

        table.align["ID"] = "c"
        table.align["Nama Pesawat"] = "l"
        table.align["Tipe"] = "l"

        for id_p, p in hasil.items():
            table.add_row([
                id_p,
                p['nama'],
                p['tipe']
            ])

        print(table)
        print(f"\nDitemukan {len(hasil)} pesawat")
    else:
        print("X Tidak ditemukan.")
```

### 3.18. Procedure Pause di `tampilan_ui.py`

```
def pause():
    input("\nTekan Enter untuk melanjutkan...")
```

### 3.19. Fungsi Utama Menu Admin di `menu.py`

```
from tampilan_ui import *
from autentikasi import *
from pesawat import *
from data_data import user_session

def menu_admin():
    while True:
        try:
            clear_screen()
            tampilkan_header(f"MENU ADMIN - {user_session['username']}")

            print("\n| _____")
            print(" |   1. Tambah Pesawat    2. Lihat Pesawat    3. Cari")
            Pesawat      | ")
            print(" |   4. Update Pesawat    5. Hapus Pesawat    6. Logout
            | ")

            print("| _____| ")
            pil = input("\n» Pilih menu: ").strip()
```

### 3.19.1. Fungsi Utama Menu Admin Tambah Pesawat di menu.py

```
if pil == "1":
    clear_screen()
    tampilkan_header("➕ TAMBAH PESAWAT")
    print()
    nama = input("📝 Nama Pesawat : ").strip()
    tipe = input("🏷️ Tipe Pesawat : ").strip()
    harga = input("💰 Harga (Rp) : ").strip()

    sukses, pesan = tambah_pesawat(nama, tipe, harga)
    print(f"\n{'✓' if sukses else '✗'} {pesan}")
    pause()
```

### 3.19.2. Fungsi Utama Menu Admin Tampilkan Daftar Pesawat di menu.py

```
elif pil == "2":
    clear_screen()
    tampilkan_header("📋 DAFTAR PESAWAT")
    print()
    tampilkan_daftar_pesawat()
    pause()
```

### 3.19.3. Fungsi Utama Menu Admin Cari Pesawat di menu.py

```
elif pil == "3":
    clear_screen()
    tampilkan_header("🔍 CARI PESAWAT")
    print()
    keyword = input("🔍 Kata kunci (nama/tipe): ").strip()
    hasil = cari_pesawat(keyword) if keyword else {}
    print()
    tampilkan_hasil_pencarian(hasil)
    pause()
```

### 3.19.4. Fungsi Utama Menu Admin Update Pesawat di menu.py

```
elif pil == "4":
    clear_screen()
    tampilan_header("📝 UPDATE PESAWAT")
    print()
    tampilan_daftar_pesawat()
    id_up = input("\n>ID Masukkan ID pesawat: ").strip()

    if id_up.isdigit():
        id_up = int(id_up)
        p = get_pesawat(id_up)

        if p:
            print("\n💡 Tekan Enter untuk skip (tidak
mengubah)")
            nama = input(f"📝 Nama [{p['nama']}]: ").strip()
            tipe = input(f"🏷️ Tipe [{p['tipe']}]: ").strip()
            harga = input(f"💰 Harga [{p['harga']}]: ").strip()
            status = input(f"📊 Status [{p['status']}]: "
").strip()

            sukses, pesan = update_pesawat(id_up, nama, tipe,
harga, status)
            print(f"\n{'✓' if sukses else '✗'} {pesan}")
        else:
            print("\n✗ ID tidak valid!")
    else:
        print("\n✗ ID harus berupa angka!")
    pause()
```

### 3.19.5. Fungsi Utama Menu Admin Hapus Pesawat di menu.py

```
elif pil == "5":
    clear_screen()
    tampilan_header("🚮 HAPUS PESAWAT")
    print()
    tampilan_daftar_pesawat()
    id_del = input("\n>ID Masukkan ID pesawat: ").strip()

    if id_del.isdigit():
        id_del = int(id_del)
        p = get_pesawat(id_del)

        if p:
            konfirmasi = input(f"\n⚠️ Hapus '{p['nama']}'?
(y/n): ").lower()
            if konfirmasi == 'y':
                sukses, pesan = hapus_pesawat(id_del)
                print(f"\n{'✓' if sukses else '✗'} {pesan}")
            else:
                print("\n✗ Penghapusan dibatalkan")
        else:
            print("\n✗ ID tidak valid!")
    else:
        print("\n✗ ID harus berupa angka!")
    pause()
```

### 3.19.6. Fungsi Utama Menu Admin Logout di menu.py

```
elif pil == "6":
    logout_user()
    print("\n👋 Logout berhasil!")
    pause()
    break

except Exception as e:
    print(f"Error: {e}")
    pause()
```

### 3.20. Fungsi Utama Menu User di menu.py

```
def menu_user():
    while True:
        try:
            clear_screen()
            tampilan_header(f"MENU PENGGUNA - {user_session['username']}")

            print("\n|_____")
            print(" | 1. Lihat Pesawat    2. Cari Pesawat    3. Logout")
            print(" |_____")
            print(" |_____| ")

            pil = input("\n» Pilih menu: ").strip()
```

#### 3.20.1. Fungsi Utama Menu User Tampilkan Daftar Pesawat di menu.py

```
if pil == "1":
    clear_screen()
    tampilan_header("DAFTAR PESAWAT")
    print()
    tampilan_daftar_pesawat()
    pause()
```

#### 3.20.2. Fungsi Utama Menu User Cari Pesawat di menu.py

```
elif pil == "2":
    clear_screen()
    tampilan_header("🔍 CARI PESAWAT")
    print()
    keyword = input("🔍 Kata kunci (nama/tipe): ").strip()
    hasil = cari_pesawat(keyword) if keyword else {}
    print()
    tampilan_hasil_pencarian_user(hasil)
    pause()
```

### 3.20.3. Fungsi Utama Menu User Logout di menu.py

```
        elif pil == "3":
            logout_user()
            print("\n👋 Logout berhasil!")
            pause()
            break

        except Exception as e:
            print(f"Error: {e}")
            pause()
```

### 3.21. Menu Utama di menu.py

```
def menu_utama():
    while True:
        try:
            clear_screen()
            tampilan_header("✈️ SISTEM MANAJEMEN DATA PESAWAT ✈️")

            print("\n| _____ |")
            print(" |     1. Login    2. Register    3. Keluar")
            print(" | _____ |")

            print("\n| _____ |")
            pil = input("\n» Pilih menu: ").strip()
```

### 3.21.1. Menu Utama Login di menu.py

```
if pil == "1":
    clear_screen()
    tampilan_header("🔒 LOGIN")
    print()
    user = input("👤 Username: ").strip()
    pwd = input("🔑 Password: ").strip()

    valid, data = validasi_login(user, pwd)
    if valid:
        login_user(user, data["role"])
        print(f"\n✅ Login berhasil! Selamat datang, {user}!")
        pause()

        if data["role"] == "admin":
            menu_admin()
        else:
            menu_user()
    else:
        print("\n❌ Login gagal! Username atau password salah.")
    pause()
```

### 3.21.2. Menu Utama Register di menu.py

```
elif pil == "2":
    clear_screen()
    tampilan_header("📝 REGISTER")
    print()
    user = input("👤 Username baru : ").strip()
    pwd = input("🔑 Password : ").strip()
    conf = input("🔑 Konfirmasi Pass : ").strip()

    sukses, pesan = register_user(user, pwd, conf)
    print(f"\n{'✅' if sukses else '❌'} {pesan}")
    pause()
```

### 3.21.3. Menu Utama Hentikan Program di menu.py

```
    elif pil == "3":
        clear_screen()
        print("\n" + "*80)
        print("✈ Terima kasih telah menggunakan Sistem Manajemen
Data Pesawat! ✈.center(80))
        print("*80 + "\n")
        break

    except Exception as e:
        print(f"Error: {e}")
        pause()
```

### 3.22. Program yang menjalankan

```
from menu import menu_utama

if __name__ == "__main__":
    try:
        menu_utama()
    except KeyboardInterrupt:
        print("\n\nProgram dihentikan.")
    except Exception as e:
        print(f"\nError: {e}")
```

#### 4. Hasil Output

```
=====
          ✈ SISTEM MANAJEMEN DATA PESAWAT ✈
=====

1. Login    2. Register    3. Keluar

» Pilih menu: █
```

*Gambar 4.1 Output program  
Menu utama*

```
=====
          🔒 LOGIN
=====

👤 Username: user1
🔑 Password: user123

✓ Login berhasil! Selamat datang, user1!

Tekan Enter untuk melanjutkan...█
```

*Gambar 4.2 Output program  
Login user*

```
=====
          MENU PENGGUNA - user1
=====

1. Lihat Pesawat    2. Cari Pesawat    3. Logout

» Pilih menu: █
```

*Gambar 4.3 Output program  
Menu user*

DAFTAR PESAWAT				
ID	Nama Pesawat	Tipe	Harga (Rp)	Status
1	Boeing 737 MAX 8	Komersial	1,250,000,000	Tersedia
2	Airbus A320neo	Komersial	1,180,000,000	Tersedia
3	Cessna Citation X	Jet Pribadi	230,000,000	Tersedia
4	Gulfstream G700	Jet Pribadi	760,000,000	Tersedia
5	ATR 72-600	Regional	250,000,000	Tersedia
6	LM F-35 Lightning II	Militer	1,850,000,000	Tersedia
7	Boeing 777-300ER	Komersial	1,800,000,000	Tersedia
8	Airbus A350-1000	Komersial	1,750,000,000	Tersedia
9	Dassault Falcon 8X	Jet Pribadi	580,000,000	Tersedia
10	Pilatus PC-24	Jet Pribadi	110,000,000	Tersedia
11	Embraer E195-E2	Regional	300,000,000	Tersedia
12	Sukhoi Superjet 100	Regional	270,000,000	Tersedia
13	Antonov An-225 Mriya	Kargo	2,500,000,000	Tersedia
14	Boeing 747-8F	Kargo	1,500,000,000	Tersedia
15	Airbus BelugaXL	Kargo	1,200,000,000	Tersedia

Total Pesawat: 15 unit | Total Nilai: Rp 15,530,000,000

Tekan Enter untuk melanjutkan...

*Gambar 4.4 Output program  
Menu user daftar pesawat*

CARI PESAWAT		
Kata kunci (nama/tipe): beluga		
ID	Nama Pesawat	Tipe
15	Airbus BelugaXL	Kargo
Ditemukan 1 pesawat		
Tekan Enter untuk melanjutkan...		

*Gambar 4.5 Output program  
Menu user cari pesawat*

```
=====
          🔒 LOGIN
=====

👤 Username: admin
🔑 Password: admin123

✓ Login berhasil! Selamat datang, admin!

Tekan Enter untuk melanjutkan... █
```

*Gambar 4.6 Output program  
Login admin*

```
=====
          MENU ADMIN - admin
=====

1. Tambah Pesawat   2. Lihat Pesawat   3. Cari Pesawat
4. Update Pesawat  5. Hapus Pesawat  6. Logout

» Pilih menu: █
```

*Gambar 4.7 Output program  
Menu admin*

```
=====
          + TAMBAH PESAWAT
=====

📅 Nama Pesawat : tralalelo tralala
🏷️ Tipe Pesawat : brainrot
💰 Harga (Rp)   : 1000000000

✓ Pesawat berhasil ditambahkan!

Tekan Enter untuk melanjutkan... █
```

*Gambar 4.8 Output program  
Menu admin tambah pesawat*

UPDATE PESAWAT				
ID	Nama Pesawat	Tipe	Harga (Rp)	Status
1	Boeing 737 MAX 8	Komersial	1,250,000,000	Tersedia
2	Airbus A320neo	Komersial	1,180,000,000	Tersedia
3	Cessna Citation X	Jet Pribadi	230,000,000	Tersedia
4	Gulfstream G700	Jet Pribadi	760,000,000	Tersedia
5	ATR 72-600	Regional	250,000,000	Tersedia
6	LM F-35 Lightning II	Militer	1,850,000,000	Tersedia
7	Boeing 777-300ER	Komersial	1,800,000,000	Tersedia
8	Airbus A350-1000	Komersial	1,750,000,000	Tersedia
9	Dassault Falcon 8X	Jet Pribadi	580,000,000	Tersedia
10	Pilatus PC-24	Jet Pribadi	110,000,000	Tersedia
11	Embraer E195-E2	Regional	300,000,000	Tersedia
12	Sukhoi Superjet 100	Regional	270,000,000	Tersedia
13	Antonov An-225 Mriya	Kargo	2,500,000,000	Tersedia
14	Boeing 747-8F	Kargo	1,500,000,000	Tersedia
15	Airbus BelugaXL	Kargo	1,200,000,000	Tersedia
16	tralalelo tralala	brainrot	100,000,000,000	Tersedia
17	tralalelo tralala	brainrot	1,000,000,000	Tersedia

Total Pesawat: 17 unit | Total Nilai: Rp 116,530,000,000

**ID** Masukkan ID pesawat: 17

- 💡 Tekan Enter untuk skip (tidak mengubah)
- 📝 Nama [tralalelo tralala]: bombardini
- 🏷️ Tipe [brainrot]: brainrot
- 💰 Harga [1000000000]: 10000000000000000000
- FLAG Status [Tersedia]: sold

Pesawat berhasil diupdate!

Tekan Enter untuk melanjutkan... ↵

*Gambar 4.9 Output program  
Menu admin update pesawat*

=====

HAPUS PESAWAT

=====

ID	Nama Pesawat	Tipe	Harga (Rp)	Status
1	Boeing 737 MAX 8	Komersial	1,250,000,000	Tersedia
2	Airbus A320neo	Komersial	1,180,000,000	Tersedia
3	Cessna Citation X	Jet Pribadi	230,000,000	Tersedia
4	Gulfstream G700	Jet Pribadi	760,000,000	Tersedia
5	ATR 72-600	Regional	250,000,000	Tersedia
6	LM F-35 Lightning II	Militer	1,850,000,000	Tersedia
7	Boeing 777-300ER	Komersial	1,800,000,000	Tersedia
8	Airbus A350-1000	Komersial	1,750,000,000	Tersedia
9	Dassault Falcon 8X	Jet Pribadi	580,000,000	Tersedia
10	Pilatus PC-24	Jet Pribadi	110,000,000	Tersedia
11	Embraer E195-E2	Regional	300,000,000	Tersedia
12	Sukhoi Superjet 100	Regional	270,000,000	Tersedia
13	Antonov An-225 Mriya	Kargo	2,500,000,000	Tersedia
14	Boeing 747-8F	Kargo	1,500,000,000	Tersedia
15	Airbus BelugaXL	Kargo	1,200,000,000	Tersedia
16	tralalelo tralala	brainrot	100,000,000,000,000	Tersedia
17	bombardini	brainrot	100,000,000,000,000,000	sold

=====

Total Pesawat: 17 unit | Total Nilai: Rp 100,000,115,530,000,000

=====

**ID** Masukkan ID pesawat: 17

**⚠** Hapus 'bombardini'? (y/n): y

**✓** Pesawat 'bombardini' berhasil dihapus!

Tekan Enter untuk melanjutkan... █

*Gambar 4.10 Output program  
Menu admin hapus pesawat*

```
=====
          REGISTER
=====

👤 Username baru      : userbaru
🔑 Password          : user123
🔑 Konfirmasi Pass   : user123

✅ Registrasi berhasil!

Tekan Enter untuk melanjutkan...|
```

Gambar 4.11 Output program  
Menu registrasi akun baru

```
=====
        Terima kasih telah menggunakan Sistem Manajemen Data Pesawat! ✈️
=====

PS C:\Users\rizal\Desktop\Praktikum_APD_B2-25> |
```

Gambar 4.12 Output akhiri program

```
=====
          LOGIN
=====

👤 Username: bukanorangsini
🔑 Password: anomali

✖️ Login gagal! Username atau password salah.

Tekan Enter untuk melanjutkan...|
```

```
=====
          REGISTER
=====

👤 Username baru      : userbaru
🔑 Password          : sengajasalah
🔑 Konfirmasi Pass   : salahsengaja

✖️ Password dan konfirmasi tidak cocok!

Tekan Enter untuk melanjutkan...|
```

Gambar 4.13 Output program  
Error handling pada  
login,, dan register akun baru

## 5. Langkah-langkah GIT

### 5.1 GIT Add

```
PS C:\Users\rizal\Desktop\Praktikum_APD_B2-25> git add .
```

Gambar 5.1

Menambahkan file yang sudah diubah ke staging area, yaitu area sementara sebelum disimpan permanen ke repository.

### 5.2 GIT Commit

```
PS C:\Users\rizal\Desktop\Praktikum_APD_B2-25> git commit -m "final"
[main a7c1f2a] final
 2 files changed, 4 insertions(+), 4 deletions(-)
```

Gambar 5.2

Menyimpan file yang sudah di-add ke dalam repository Git dengan pesan tertentu. Pesan commit biasanya menjelaskan perubahan yang dilakukan.

### 5.3 GIT Push

```
PS C:\Users\rizal\Desktop\Praktikum_APD_B2-25> git push
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 781 bytes | 781.00 KiB/s, done.
Total 8 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 6 local objects.
To https://github.com/print-rizalll/Praktikum_APD_B2-25.git
  992d36c..a7c1f2a main -> main
PS C:\Users\rizal\Desktop\Praktikum_APD_B2-25>
```

Gambar 5.3

Mengirim (upload) commit dari repository lokal ke repository remote (GitHub).