

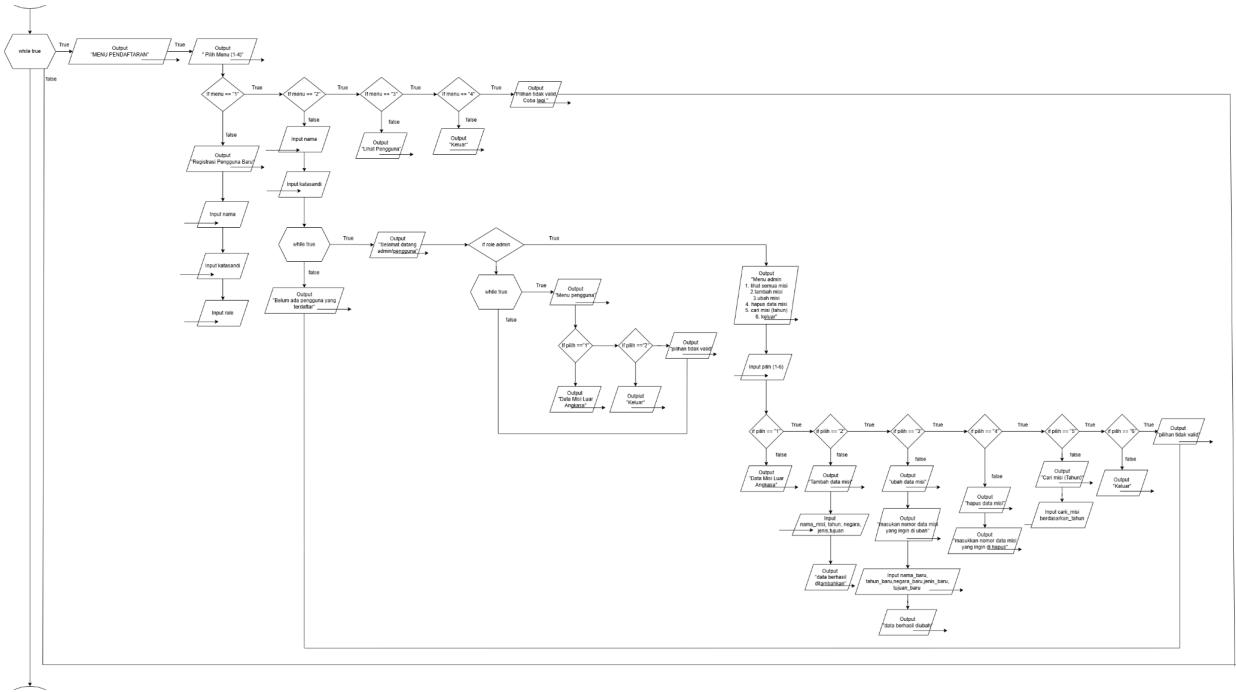
**LAPORAN PRAKTIKUM  
POSTTEST (8)  
ALGORITMA PEMROGRAMAN DASAR**



**Disusun oleh:  
Muhamad Akbar Pratama (2509106091)  
Kelas (C1 '25)**

**PROGRAM STUDI INFORMATIKA  
UNIVERSITAS MULAWARMAN SAMARINDA  
2025**

## 1. Flowchart



## Gambar 1.1 Flowchart

## **2. Deskripsi Singkat Program**

Program ini digunakan untuk mengelola data misi luar angkasa dengan menggunakan fungsi dengan parameter dan tanpa parameter, prosedur, variabel global, variabel lokal. Program juga menyediakan fitur registrasi dan login, serta menampilkan daftar pengguna yang terdaftar. Fungsi role pada masing masing yaitu:

- **Admin** bisa melihat, menambah, mengubah, menghapus, dan mencari misi berdasarkan tahun.
  - **Pengguna** biasa hanya bisa melihat daftar misi.

Tujuannya adalah untuk mensimulasikan sistem manajemen data sederhana dan pengelolaan informasi misi luar angkasa.

### 3. Source Code

#### A. Isi Data Misi Luar Angkasa

```
from prettytable import PrettyTable

misi_luar_angkasa = [
    {"nama": "Voyager 1", "tahun": 1977, "negara": "Amerika Serikat",
     "jenis": "Probe", "tujuan": "Eksplorasi ruang antar bintang"},
    {"nama": "Apollo 11", "tahun": 1969, "negara": "Amerika Serikat",
     "jenis": "Misi Berawak", "tujuan": "Pendaratan di Bulan"},
    {"nama": "Tianwen-1", "tahun": 2020, "negara": "China", "jenis": "Rover",
     "tujuan": "Eksplorasi Mars"},
    {"nama": "Rosetta", "tahun": 2004, "negara": "Eropa", "jenis": "Probe",
     "tujuan": "Meneliti Komet 67P"},
    {"nama": "James Webb", "tahun": 2021, "negara": "Internasional",
     "jenis": "Teleskop", "tujuan": "Mengamati galaksi dan bintang jauh"}
]

def hitung_jumlah_misi():
    return len(misi_luar_angkasa)

def cari_misi_berdasarkan_tahun(tahun):
    return [m for m in misi_luar_angkasa if m["tahun"] == tahun]

def tampilkan_semua_misi():
    if not misi_luar_angkasa:
        print("Belum ada data misi.")
        return

    table = PrettyTable()
    table.field_names = ["No", "Nama", "Tahun", "Negara", "Jenis", "Tujuan"]

    for i, m in enumerate(misi_luar_angkasa, start=1):
        table.add_row([i, m["nama"], m["tahun"], m["negara"], m["jenis"],
                      m["tujuan"]])

    print(table)
    print(f"Total misi: {hitung_jumlah_misi()}")

def tambah_misi():
    try:
        nama = input("Nama Misi: ")
        tahun = int(input("Tahun: "))
        negara = input("Negara: ")
        jenis = input("Jenis: ")
        tujuan = input("Tujuan: ")
    
```

```

misi_luar_angkasa.append({
    "nama": nama,
    "tahun": tahun,
    "negara": negara,
    "jenis": jenis,
    "tujuan": tujuan
})
print("Misi berhasil ditambahkan!")
except ValueError:
    print("Tahun harus berupa angka!")

def ubah_misi():
    tampilan_semua_misi()
    try:
        idx = int(input("Masukkan nomor misi yang ingin diubah: ")) - 1
        if 0 <= idx < len(misi_luar_angkasa):
            m = misi_luar_angkasa[idx]
            m["nama"] = input(f"Nama baru ({m['nama']}): ") or m["nama"]
            m["tahun"] = int(input(f"Tahun baru ({m['tahun']}): ") or
m["tahun"])
            m["negara"] = input(f"Negara baru ({m['negara']}): ") or
m["negara"]
            m["jenis"] = input(f"Jenis baru ({m['jenis']}): ") or m["jenis"]
            m["tujuan"] = input(f"Tujuan baru ({m['tujuan']}): ") or
m["tujuan"]
            print("Data berhasil diperbarui!")
        else:
            print("Nomor tidak valid!")
    except ValueError:
        print("Input harus berupa angka!")

def hapus_misi():
    tampilan_semua_misi()
    try:
        idx = int(input("Masukkan nomor misi yang ingin dihapus: ")) - 1
        if 0 <= idx < len(misi_luar_angkasa):
            del misi_luar_angkasa[idx]
            print("Misi berhasil dihapus!")
        else:
            print("Nomor tidak valid!")
    except ValueError:
        print("Input harus berupa angka!")

```

## B. Data Login

```
pengguna = [("Akbar", "123", "admin")]
login_user = None

def registrasi():
    global pengguna
    nama = input("Masukkan nama pengguna: ")
    sandi = input("Masukkan kata sandi: ")
    role = input("Masukkan role (admin/pengguna): ")

    if any(p[0] == nama for p in pengguna):
        print("Nama sudah terdaftar!")
    elif role not in ("admin", "pengguna"):
        print("Role tidak valid!")
    else:
        pengguna.append((nama, sandi, role))
        print("Registrasi berhasil!")

def login():
    global login_user
    nama = input("Masukkan nama pengguna: ")
    sandi = input("Masukkan kata sandi: ")
    login_user = next((p for p in pengguna if p[0] == nama and p[1] == sandi), None)

    if login_user:
        print(f"Selamat datang, {login_user[0]}! (Role: {login_user[2]})")
    else:
        print("Login gagal!")
    return login_user

def tampilkan_pengguna():
    from prettytable import PrettyTable

    if not pengguna:
        print("Belum ada pengguna.")
        return

    table = PrettyTable()
    table.field_names = ["Nama", "Role"]
    for p in pengguna:
        table.add_row([p[0], p[2]])
    print(table)
```

## C. Main

```
from Login_Misi import registrasi, login, tampilan_pengguna, login_user
from Data_Misi import tampilan_semua_misi, tambah_misi, ubah_misi,
hapus_misi, cari_misi_berdasarkan_tahun

while True:
    print("""
=====
|           MENU UTAMA          |
=====
| 1. Registrasi                |
| 2. Login                      |
| 3. Lihat Pengguna            |
| 4. Keluar                     |
=====
""")
```

"")

```
    menu = input("Pilih menu (1-4): ")

    if menu == "1":
        registrasi()

    elif menu == "2":
        user = login()
        if user:
            if user[2] == "admin":
                while True:
                    print("""
=====
|           MENU ADMIN          |
=====
| 1. Lihat Semua Misi         |
| 2. Tambah Misi Baru         |
| 3. Ubah Data Misi           |
| 4. Hapus Data Misi          |
| 5. Cari Misi (Tahun)        |
| 6. Logout                    |
=====
""")
```

"")

```
                pilih = input("Pilih menu (1-6): ")

                if pilih == "1":
                    tampilan_semua_misi()
                elif pilih == "2":
                    tambah_misi()
                elif pilih == "3":
                    ubah_misi()
                elif pilih == "4":
                    hapus_misi()
                elif pilih == "5":
```

```

try:
    tahun = int(input("Masukkan tahun: "))
    hasil = cari_misi_berdasarkan_tahun(tahun)
    if hasil:
        from prettytable import PrettyTable
        table = PrettyTable()
        table.field_names = ["Nama", "Negara",
        "Tahun"]
        for m in hasil:
            table.add_row([m["nama"], m["negara"],
            m["tahun"]])
        print(table)
    else:
        print("Tidak ada misi pada tahun tersebut.")
except ValueError:
    print("Tahun harus angka!")
elif pilih == "6":
    break
else:
    print("Pilihan tidak valid!")

else:
    while True:
        print("""
=====
|      MENU PENGGUNA      |
=====
| 1. Lihat Data Misi   |
| 2. Logout             |
=====
""")
```

"")

```

        pilih = input("Pilih menu (1-2): ")

        if pilih == "1":
            tampilkan_semua_misi()
        elif pilih == "2":
            break
        else:
            print("Pilihan tidak valid!")

elif menu == "3":
    tampilkan_pengguna()

elif menu == "4":
    print("Keluar dari program...")
    break

else:
    print("Pilihan tidak valid!")
```

## 4. Hasil Output

### A. Menu Pendaftaran dan Login Admin

```
=====  
|           MENU PEMDAFTARAN          |  
=====  
| 1. Registrasi  
| 2. Login  
| 3. Lihat Pengguna  
| 4. Keluar  
=====  
  
Pilih menu (1-4): 2  
Masukkan nama pengguna: Akbar  
Masukkan kata sandi: 123  
Selamat datang, Akbar! Role: admin
```

Gambar 4.1 Output Login

```
== DAFTAR MISI LUAR ANGKASA ==  
1. Nama: Voyager 1 | Tahun: 1977 | Negara: Amerika Serikat | Jenis: Probe | Tujuan: Eksplorasi ruang antar bintang  
2. Nama: Apollo 11 | Tahun: 1969 | Negara: Amerika Serikat | Jenis: Misi Berawak | Tujuan: Pendaratan di Bulan  
3. Nama: Tianwen-1 | Tahun: 2020 | Negara: China | Jenis: Rover | Tujuan: Eksplorasi Mars  
4. Nama: Rosetta | Tahun: 2004 | Negara: Eropa | Jenis: Probe | Tujuan: Meneliti Komet 67P  
5. Nama: James Webb | Tahun: 2021 | Negara: Internasional | Jenis: Teleskop | Tujuan: Mengamati galaksi dan bintang jauh  
  
Total misi: 5
```

Gambar 4.2 Output Menu

```
=====
|      MENU ADMIN      |
=====

| 1. Lihat Semua Misi   |
| 2. Tambah Misi Baru   |
| 3. Ubah Data Misi     |
| 4. Hapus Data Misi    |
| 5. Cari Misi (Tahun)  |
| 6. Logout              |
=====

Pilih menu (1-6): 2
Nama Misi: Artemis
Tahun: 2000
Negara: indonesia
Jenis Misi: berawak
Tujuan Misi: mars
Misi baru berhasil ditambahkan!
```

Gambar 4.3 Output Menu 2

```
Pilih menu (1-6): 3
===
== DAFTAR MISI LUAR ANGKASA ==
1. Nama: Voyager 1 | Tahun: 1977 | Negara: Amerika Serikat | Jenis: Probe | Tujuan: Eksplorasi ruang antar bintang
2. Nama: Apollo 11 | Tahun: 1969 | Negara: Amerika Serikat | Jenis: Misi Berawak | Tujuan: Pendaratan di Bulan
3. Nama: Tianwen-1 | Tahun: 2020 | Negara: China | Jenis: Rover | Tujuan: Eksplorasi Mars
4. Nama: Rosetta | Tahun: 2004 | Negara: Eropa | Jenis: Probe | Tujuan: Meneliti Komet 67P
5. Nama: James Webb | Tahun: 2021 | Negara: Internasional | Jenis: Teleskop | Tujuan: Mengamati galaksi dan bintang jauh
6. Nama: Artemis | Tahun: 2000 | Negara: indonesia | Jenis: berawak | Tujuan: mars

Masukkan nomor misi yang ingin diubah: 6
Data lama: {'nama': 'Artemis', 'tahun': 2000, 'negara': 'indonesia', 'jenis': 'berawak', 'tujuan': 'mars'}
Nama baru (Artemis): Artemis 1
Tahun baru (2000): 2022
Negara baru (indonesia): Amerika Serikat
Jenis baru (berawak): Uji coba berawak
Tujuan baru (mars): Bulan
Data misi berhasil diperbarui!
```

Gambar 4.4 Output Menu 3

```
Pilih menu (1-6): 4

==== DAFTAR MISI LUAR ANGKASA ====
1. Nama: Voyager 1 | Tahun: 1977 | Negara: Amerika Serikat | Jenis: Probe | Tujuan: Eksplorasi ruang antar bintang
2. Nama: Apollo 11 | Tahun: 1969 | Negara: Amerika Serikat | Jenis: Misi Berawak | Tujuan: Pendaratan di Bulan
3. Nama: Tianwen-1 | Tahun: 2020 | Negara: China | Jenis: Rover | Tujuan: Eksplorasi Mars
4. Nama: Rosetta | Tahun: 2004 | Negara: Eropa | Jenis: Probe | Tujuan: Meneliti Komet 67P
5. Nama: James Webb | Tahun: 2021 | Negara: Internasional | Jenis: Teleskop | Tujuan: Mengamati galaksi dan bintang jauh
6. Nama: Artemis 1 | Tahun: 2022 | Negara: Amerika Serikat | Jenis: Uji coba berawak | Tujuan: Bulan

Masukkan nomor misi yang ingin dihapus: 6
Data misi berhasil dihapus!
```

Gambar 4.4 Output Menu 4

```
=====
|           MENU ADMIN          |
=====
| 1. Lihat Semua Misi      |
| 2. Tambah Misi Baru       |
| 3. Ubah Data Misi         |
| 4. Hapus Data Misi        |
| 5. Cari Misi (Tahun)      |
| 6. Logout                  |
=====

Pilih menu (1-6): 5
Masukkan tahun misi: 2020

Hasil pencarian:
Nama: Tianwen-1 | Negara: China
```

Gambar 4.5 Output Menu 5

## C. Menu Pengguna

```
=====
|       MENU PENGGUNA      |
=====
| 1. Lihat Data Misi     |
| 2. Logout               |
=====

Pilih menu (1-2): 1

== DAFTAR MISI LUAR ANGKASA ==
1. Nama: Voyager 1 | Tahun: 1977 | Negara: Amerika Serikat | Jenis: Probe | Tujuan: Eksplorasi ruang antar bintang
2. Nama: Apollo 11 | Tahun: 1969 | Negara: Amerika Serikat | Jenis: Misi Berawak | Tujuan: Pendaratan di Bulan
3. Nama: Tianwen-1 | Tahun: 2020 | Negara: China | Jenis: Rover | Tujuan: Eksplorasi Mars
4. Nama: Rosetta | Tahun: 2004 | Negara: Eropa | Jenis: Probe | Tujuan: Meneliti Komet 67P
5. Nama: James Webb | Tahun: 2021 | Negara: Internasional | Jenis: Teleskop | Tujuan: Mengamati galaksi dan bintang jauh
```

Gambar 4.6 Output Menu Pengguna

## 5. Langkah-langkah GIT

### 5.1 GIT ADD

Fungsi git add ini digunakan untuk menambahkan semua file yang ada di folder saat ini

```
PS C:\Users\muham\Downloads\Praktikum-Apd> Git add .
```

Gambar 5.1 git add

### 5.2 GIT COMMIT

Fungsinya mengkonfirmasi perubahan atau penambahan file baru

```
PS C:\Users\muham\Downloads\Praktikum-Apd> git commit -m "commit"
```

Gambar 5.2 git commit

### 5.3 GIT PUSH

Fungsinya mengunggah file dari repository local ke github

```
PS C:\Users\muham\Downloads\Praktikum-Apd> git push -u origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 223.77 KiB | 10.17 MiB/s, done.
Total 8 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Noname240806/Praktikum-Apd-C1.git
  911d72e..a01ee8d  main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5.3 git push