

LAPORAN PRAKTIKUM POSTTEST 7
ALGORITMA DAN PEMROGRAMAN DASAR



Richo Anan Rizky Putra B1

2409106062

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

2024

LATAR BELAKANG

Pada posttest ke-7 ini, praktikan diminta untuk melanjutkan program yang telah dibuat pada Posttest 6 dengan ketentuan:

- Buat 3 fungsi dengan dan tanpa parameter
- Buat 2 prosedur
- Gunakan minimal 3 variable global dan 5 variable local

Dan nilai tambahan apabila Praktikan dapat:

- Menggunakan fungsi rekursif, dan
- Error Handling

Program ini di tambahkan beberapa fungsi-fungsi dan prosedur berdasarkan ketentuan yang diinstruksikan kepada Praktikan.

Dalam menambahkan fungsi-fungsi dan prosedur tersebut, saya mengalami kendala, yaitu:

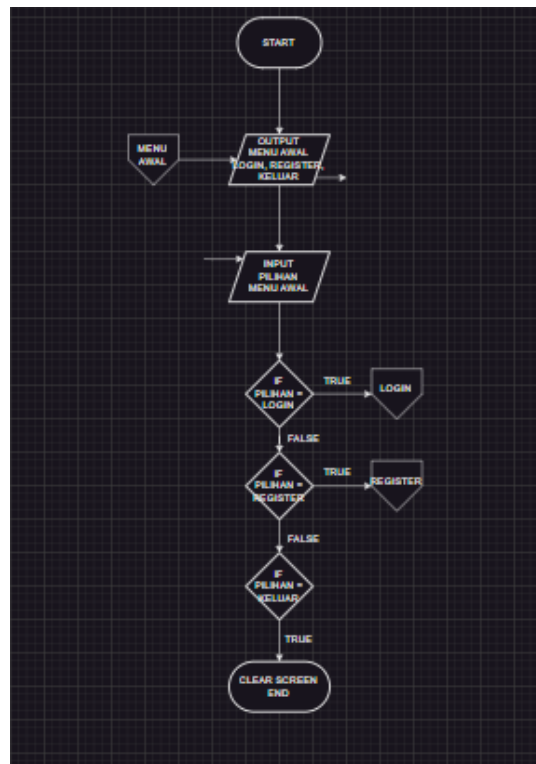
-Pembuatan fungsi rekursif, dikarenakan kurangnya info yang saya dapatkan dalam pengerjaannya

Saya sebagai Praktikan menyelesaikan program ini menggunakan ***draw.io*** untuk membuat ***Flowchart*** dan menggunakan ***Visual Studio Code*** untuk membuat **kodingannya**. Saya membagi file kodingannya menjadi 4 file, tapi tetap dalam folder yang sama, dengan nama file:

1. *main.py*, sebagai file untuk memulai program yang akan dilanjut ke file *user.py*.
2. *user.py*, sebagai file yang berisi menu awal untuk melakukan *login* sebagai *admin* atau *user* biasa, maupun langsung keluar dari program; file ini juga berisi menu utama yang berisi beberapa pilihan berdasarkan role dari login
3. *atlet.py*, yang berisi kodingan tentang isi dari fungsi yang di dalam menu utama yaitu: tambah atlet, lihat daftar atlet, edit data atlet yang ada di daftar, dan terakhir menghapus data atlet dari daftar; dan
4. *utility.py*, yang berisi kodingan pendukung untuk file-file yang lain, yaitu: *clear_screen*, dan validasi input.

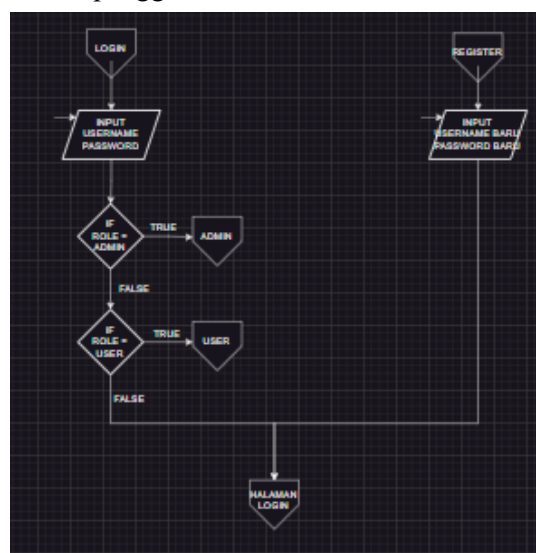
SOLUSI

Flowchart

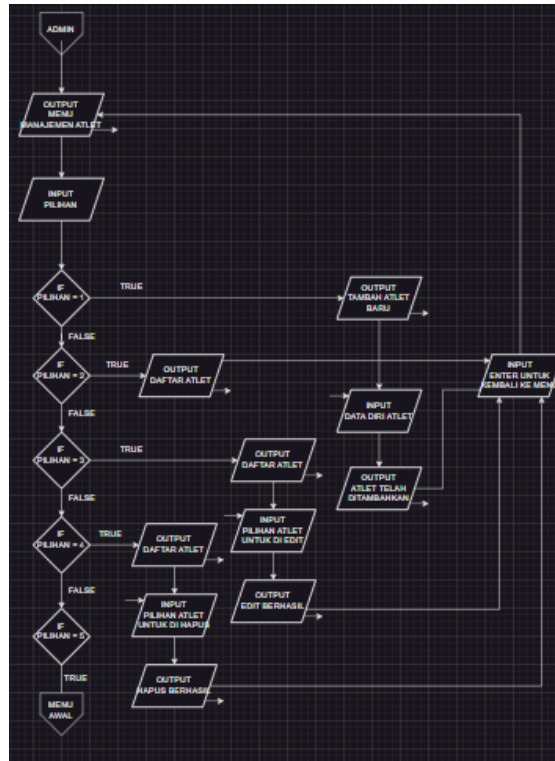


Gambar 1.1

Awal dari *flowchart*, pengguna akan ditampilkan menu awal yang berisi: *login*, *register*, dan *keluar*. Jika pengguna memilih *login* akan diminta untuk memasukkan *username* dan *password*. Jika pengguna memilih *register* maka akan diminta untuk memasukkan *username* dan *password* yang baru dan kemudian Kembali ke menu awal untuk melakukan login. Jika pengguna memilih *keluar* maka akan mengakhiri program.



Gambar 1.2



Gambar 1.3

Jika pengguna langsung melakukan *login* tanpa *register* maka akan mendapat role *admin*, dan akan ditampilkan menu utama yang berisi 5 pilihan, yaitu:

1. Tambah atlet, 2. Lihat atlet, 3. Edit atlet, 4. Hapus atlet, dan 5. Keluar.

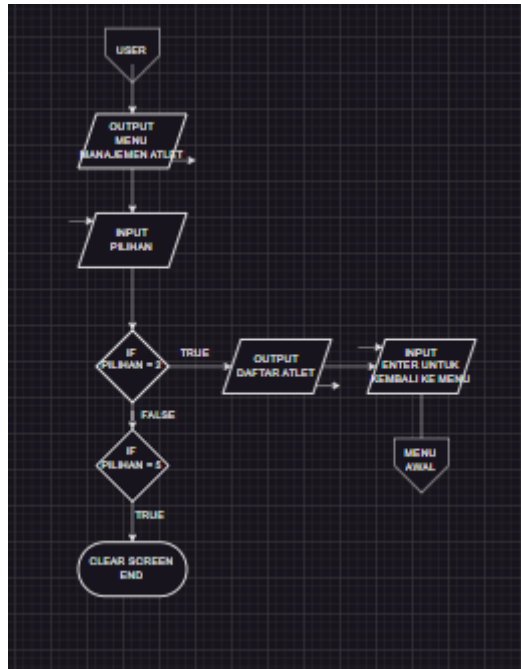
Jika *admin* memilih 1, akan diminta untuk memasukkan data diri atlet yang terdiri dari: Nama, Usia, Cabor, Posisi, Statistik, dan Prestasi.

Jika *admin* memilih 2, akan diperlihatkan daftar atlet yang ada di daftar

Jika *admin* memilih 3, akan diperlihatkan daftar atlet yang ada di daftar, kemudian akan memilih data mana yang ingin di edit, antara itu Nama, Usia, Cabor, Posisi, Statistik, ataupun Prestasi.

Jika *admin* memilih 4, akan diperlihatkan daftar atlet yang ada di daftar, kemudian akan memilih data mana yang ingin dihapus.

Jika *admin* memilih 5, akan menutup Menu Utama dan akan kembali ke Menu Awal yang ada pada Gambar 1.1.



Gambar 1.4

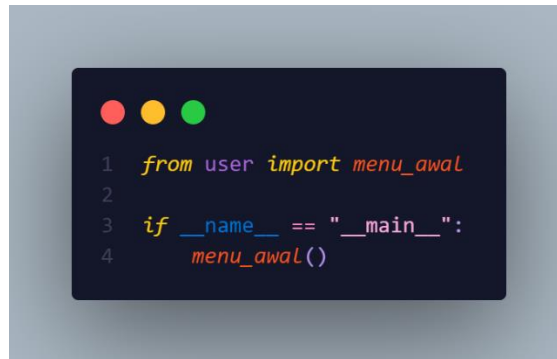
Jika pengguna melakukan *register* terlebih dahulu lalu melakukan *login*, maka akan mendapat role *user*, dan akan ditampilkan menu utama yang hanya berisi 2 pilihan, yaitu: 2. Lihat atlet, dan 5. Keluar.

Jika *user* memilih 2, akan diperlihatkan daftar atlet yang ada di daftar

Jika *user* memilih 5, akan menutup Menu Utama dan akan kembali ke Menu Awal yang ada pada *Gambar 1.1*.

Codingan program di Python

Seperti yang telah saya sampaikan diatas bahwa kodingan saya bagi menjadi 4 file dan dalam 1 folder yang sama,yaitu: *main.py*, *user.py*, *atlet.py*, dan *utility.py*; dan akan saya jelaskan sedikit

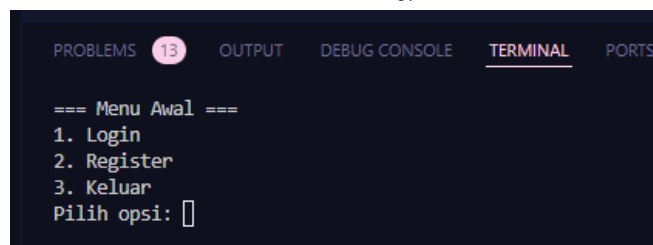


Gambar 2.1 *main.py*

Gambar 2.1 merupakan kodingan file *main.py*, yang merupakan tempat program akan dijalankan dengan me-run, lalu akan ditampilkan menu awal seperti *Gambar 2.3*



Gambar 2.2 *user.py*



Gambar 2.3

Awal dari program, pengguna akan ditampilkan menu awal yang berisi: *login*, *register*, dan *keluar*. Jika pengguna memilih *login* akan diminta untuk memasukkan *username* dan *password*, seperti yang ditampilkan pada *Gambar 2.6*. Jika pengguna memilih *register* maka akan diminta untuk memasukkan *username* dan *password* yang baru dan akan ditampilkan pada *Gambar 2.7* dan kemudian Kembali ke menu awal untuk melakukan

login seperti *Gambar 2.6*. Jika pengguna memilih *keluar* maka akan mengakhiri program dan akan diperlihatkan *Gambar 2.8*.

```
1 import os
2 from utility import clear_screen, validasi_input
3 from atlet import tambah_atlet, lihat_atlet, edit_atlet, hapus_atlet
4
5 users = [{
6     "user" : "admin",
7     "pw" : "admin123",
8     "role" : "admin"
9 }]
```

Gambar 2.4 *user.py*

```
1 # Fungsi untuk Login
2 def Login():
3     clear_screen()
4     print("--- Login ---")
5     username = input("Username: ")
6     password = input("Password: ")
7
8     # Cek user di dalam database users
9     for user in users:
10         if user["user"] == username and user["pw"] == password:
11             print(f"\nSelamat datang, {username}!")
12             return user["role"]
13
14     print("\nUsername atau password salah!")
15     input("Tekan Enter untuk kembali ke login...")
16     return None
17
18 # Fungsi untuk membatasi Login
19 def coba_login(coba=1):
20     role = Login()
21     if role:
22         return role
23     elif coba < 3:
24         print(f"Percobaan login ke-{coba}. Coba lagi.")
25         return coba_login(coba + 1)
26     else:
27         print("Batas percobaan login habis!")
28         return None
```

Gambar 2.5 kodingan login *user.py*

Kodingan ini memeriksa apakah pengguna melakukan login sesuai dengan data admin yang ada. Jika melakukan kesalahan sebanyak lebih dari 3 kali maka akan Kembali ke tampilan menu awal (*Gambar 2.3*)

```
PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL PORTS

--- Login ---
Username:
Password: 
```

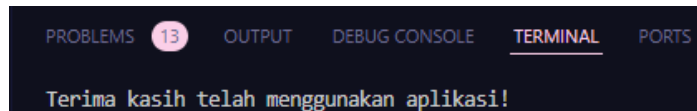
Gambar 2.6



```
PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL PORTS

--- Register ---
Masukkan username baru:
Masukkan password baru: |
```

Gambar 2.7



```
PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Terima kasih telah menggunakan aplikasi!
```

Gambar 2.8



```
1 # Fungsi untuk register
2 def register():
3     clear_screen()
4     print("--- Register ---")
5     username = input("Masukkan username baru: ")
6     password = input("Masukkan password baru: ")
7
8     # Cek apakah username sudah ada
9     for user in users:
10         if user["user"] == username:
11             print("\nUsername sudah ada!")
12             input("Tekan Enter untuk kembali ke menu...")
13             return
14
15     users.append({"user": username, "pw": password, "role": "user"})
16     print(f"\nPengguna {username} berhasil didaftarkan!")
17     input("Tekan Enter untuk kembali ke menu...")
18
```

Gambar 2.9 Kodingan registrasi *user.py*

Pada kodingan ini akan ditampilkan *Gambar 2.7*, Dimana pengguna akan melakukan register dengan memasukkan username dan password yang baru, kemudian akan Kembali ke tampilan Menu Awal *Gambar 2.3* dan melakukan login dengan memasukkan username dan password yang baru.


```
1 # Fungsi utama untuk menampilkan menu utama
2 def menu_utama(role):
3     while True:
4         clear_screen()
5         print("=== Menu Manajemen Atlet ===")
6         print("1. Tambah Atlet") if role == 'admin' else None
7         print("2. Lihat Daftar Atlet")
8         print("3. Edit Informasi Atlet") if role == 'admin' else None
9         print("4. Hapus Atlet") if role == 'admin' else None
10        print("5. Keluar")
11
12    try:
13        pilihan = validasi_input("Pilih opsi: ", int)
14        if role == "admin":
15            if pilihan == 1:
16                tambah_atlet()
17            elif pilihan == 2:
18                lihat_atlet()
19            elif pilihan == 3:
20                edit_atlet()
21            elif pilihan == 4:
22                hapus_atlet()
23            elif pilihan == 5:
24                clear_screen()
25                print("Terima kasih telah menggunakan aplikasi!")
26                break
27        else:
28            print("\nPilihan tidak valid.")
29    except: # Jika pengguna adalah 'user'
30        if pilihan == 2:
31            lihat_atlet()
32        elif pilihan == 5:
33            clear_screen()
34            print("Terima kasih telah menggunakan aplikasi!")
35            break
36        else:
37            print("\nPilihan tidak valid.")
38    except ValueError:
39        print("\nInput tidak valid. Harap masukkan angka.")
40
```

Gambar 2.10 Kodingan menu utama *user.py*

Kodingan Menu Utama akan ditampilkan kepada pengguna berdasarkan role yang didapat. Jika role pengguna admin, maka akan menampilkan *Gambar 2.10.a* dan pengguna dapat memilih antara 5 pilihan yang ada.

Jika role pengguna user, maka akan menampilkan *Gambar 2.10.b* dan pengguna hanya dapat memilih antara 2 pilihan saja

```
PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL PORTS

=== Menu Manajemen Atlet ===
1. Tambah Atlet
2. Lihat Daftar Atlet
3. Edit Informasi Atlet
4. Hapus Atlet
5. Keluar
Pilih opsi: |
```

Gambar 2.10.a

```
PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL PORTS

=== Menu Manajemen Atlet ===
2. Lihat Daftar Atlet
5. Keluar
Pilih opsi: |
```

Gambar 2.10.b

```

1  import os
2
3  # Fungsi untuk membersihkan Layar
4  def clear_screen():
5      os.system('cls')
6
7
8  # Fungsi untuk validasi input tipe data
9  def validasi_input(input_prompt, input_type):
10     while True:
11         try:
12             return input_type(input(input_prompt))
13         except ValueError:
14             print(f"Input harus berupa {input_type.__name__}. Coba lagi.")

```

Gambar 2.11 *utility.py*

Merupakan kodingan untuk utility atau fungsi-fungsi yang penting, yaitu:

- clear_screen, untuk membersihkan Terminal
- validasi_input, untuk mengecek apakah input yang dimasukkan sesuai

```

1  import os
2  from utility import clear_screen, validasi_input
3
4  atlet_list = [
5      {
6          "nama": "Erling Haaland",
7          "usia": 24,
8          "cabor": "Sepak Bola",
9          "posisi": "Penyerang",
10         "statistik": "27 Gol, 5 Assist",
11         "prestasi": "Golden Boy 2020"
12     },
13     {
14         "nama": "Stephen Curry",
15         "usia": 36,
16         "cabor": "Basket",
17         "posisi": "Point Guard",
18         "statistik": "26.4 Poin, 4.5 Rebound",
19         "prestasi": "MVP NBA 2022"
20     }
21 ]

```

Gambar 2.12 *atlet.py*

Merupakan dictionary dari list atau daftar atlet yang sudah ada

```
1 # Fungsi untuk mengedit informasi atlet (hanya admin)
2 def edit_atlet():
3     clear_screen()
4     lihat_atlet()
5     if len(atlet_list) > 0:
6         try:
7             pilihan = validasi_input("Pilih nomor atlet yang ingin diedit: ", int)
8             if 1 <= pilihan <= len(atlet_list):
9                 atlet = atlet_list[pilihan - 1]
10                print(f"\n--- Edit Informasi Atlet {atlet['nama']} ---")
11                atlet["nama"] = input(f>Nama ({atlet['nama']}): ") or atlet["nama"]
12                atlet["usia"] = validasi_input(f"Usia ({atlet['usia']}): ", int) or atlet["usia"]
13                atlet["cabor"] = input(f"Cabang ({atlet['cabor']}): ") or atlet["cabor"]
14                atlet["posisi"] = input(f"Posisi ({atlet['posisi']}): ") or atlet["posisi"]
15                atlet["statistik"] = input(f"Statistik ({atlet['statistik']}): ") or atlet["statistik"]
16                atlet["prestasi"] = input(f"Prestasi ({atlet['prestasi']}): ") or atlet["prestasi"]
17                print(f"\nAtlet {atlet['nama']} berhasil diupdate!\n")
18            else:
19                print("\nPilihan tidak valid!")
20        except ValueError:
21            print("\nInput tidak valid. Harap masukkan nomor yang benar.")
22        input("\nTekan Enter untuk kembali ke menu...")
23
```

Gambar 2.13 *atlet.py*

Pengguna akan ditampilkan daftar/list dari atlet dan akan memilih atlet mana yang ingin di edit data dirinya. Jika telah selesai, maka pengguna akan Kembali ke tampilan Menu Utama.

```
1 # Fungsi untuk menghapus atlet (hanya admin)
2 def hapus_atlet():
3     clear_screen()
4     lihat_atlet()
5
6     if len(atlet_list) > 0:
7         try:
8             pilihan = validasi_input("Pilih nomor atlet yang ingin dihapus: ", int)
9             if 1 <= pilihan <= len(atlet_list):
10                 atlet = atlet_list.pop(pilihan - 1)
11                 print(f"\nAtlet {atlet['nama']} berhasil dihapus!\n")
12             else:
13                 print("\nPilihan tidak valid!")
14         except ValueError:
15             print("\nInput tidak valid. Harap masukkan nomor yang benar.")
16         input("\nTekan Enter untuk kembali ke menu...")
```

Gambar 2.14

Pengguna akan ditampilkan daftar/list dari atlet dan akan memilih atlet mana yang ingin di hapus dari data. Jika telah selesai, maka pengguna akan Kembali ke tampilan Menu Utama.

```

1  # Fungsi untuk melihat daftar atlet (nama - usia - cabor - posisi - statistik - prestasi)
2  def Lihat_atlet():
3      clear_screen()
4      if Len(atlet_list) == 0:
5          print("\nTidak ada atlet yang terdaftar.")
6      else:
7          print("\n-- Daftar Atlet --")
8          for index, atlet in enumerate(atlet_list):
9              print(f"Atlet {index + 1}")
10             print(f"    Nama    \t:{atlet['nama']}")
11             print(f"    Usia    \t:{atlet['usia']}")
12             print(f"    Cabang  \t:{atlet['cabor']}")
13             print(f"    Posisi  \t:{atlet['posisi']}")
14             print(f"    Statistik\t:{atlet['statistik']}")
15             print(f"    Prestasi \t:{atlet['prestasi']}")
16             print("-" * 30)
17         input("\nTekan Enter untuk kembali ke menu...")
18

```

Gambar 2.15 *atlet.py*

Pengguna akan diperlihatkan daftar/list dari atlet” yang telah ada di data. Jika telah selesai, maka pengguna akan kembali ke tampilan Menu Utama.

```

1  # Fungsi untuk menambahkan atlet (hanya admin)
2  def tambah_atlet():
3      clear_screen()
4      print("\n--- Tambah Atlet Baru ---")
5      try:
6          nama = input("Masukkan Nama Atlet: ")
7          usia = validasi_input("Masukkan Usia Atlet: ", int)
8          cabang = input("Masukkan Cabang Olahraga: ")
9          posisi = input("Masukkan Posisi Atlet: ")
10         statistik = input("Masukkan Statistik: ")
11         prestasi = input("Masukkan Prestasi: ")
12
13         # Data atlet sebagai dictionary
14         atlet = {
15             "nama" : nama,
16             "usia" : usia,
17             "cabor" : cabang,
18             "posisi" : posisi,
19             "statistik" : statistik,
20             "prestasi" : prestasi
21         }
22
23         # Menambahkan atlet ke dalam dictionary
24         atlet_list.append(atlet)
25         print(f"\nAtlet {nama} berhasil ditambahkan!\n")
26     except ValueError:
27         print("\nInput usia harus berupa angka!")
28
29     input("Tekan Enter untuk kembali ke menu...")

```

Gambar 2.16 *atlet.py*

Pengguna akan diminta untuk memasukkan biodata diri dari atlet yang ingin ditambahkan, seperti: Nama, Usia, Cabang Olahraga(Cabor), Posisi, Statistik, dan Prestasi. Jika telah selesai memasukkan semua data, maka pengguna akan Kembali ke tampilan Menu Utama.