

LAPORAN PRAKTIKUM POSTTEST 5
ALGORITMA DAN PEMROGRAMAN DASAR



Richo Anan Rizky Putra B1

2409106062

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

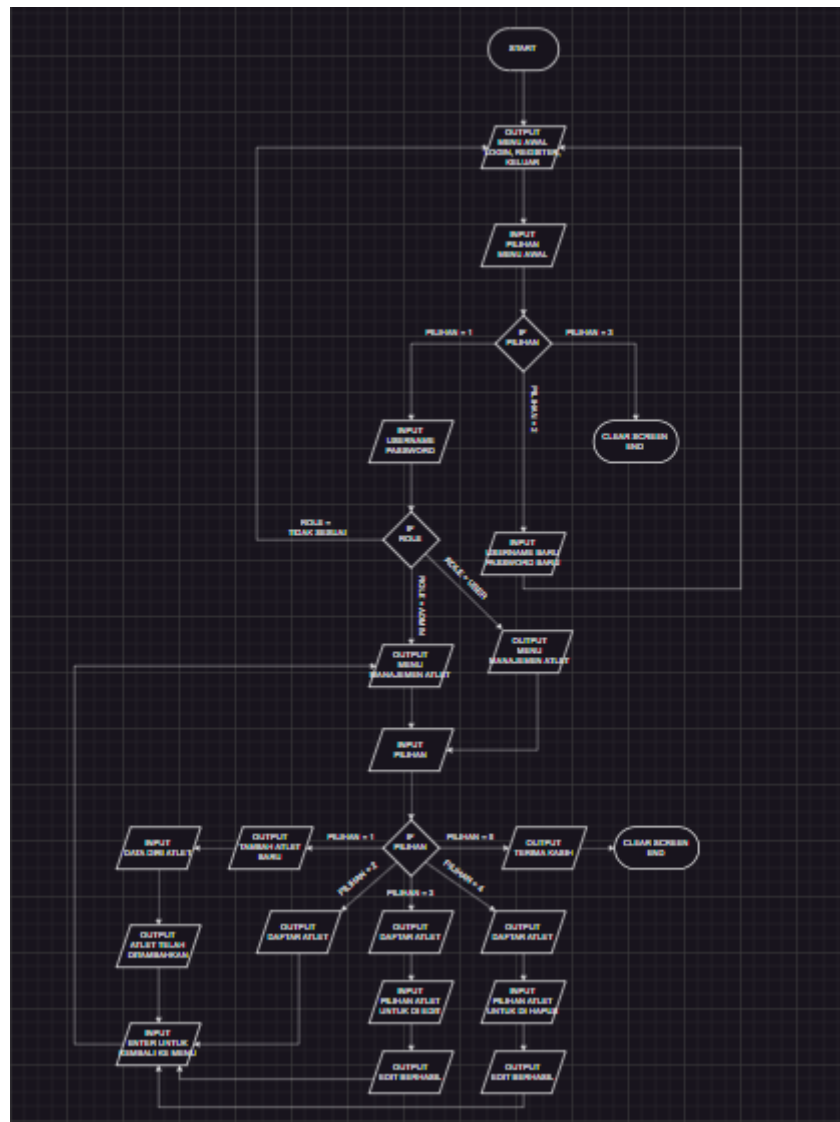
2024

LATAR BELAKANG

Pada posttest ke-5 ini, praktikan diminta untuk membuat sebuah program dengan tema yang dipilih oleh praktikan sendiri, yaitu **Manajemen Atlet**. Di dalam program tersebut awalnya pengguna akan diminta untuk melakukan login admin (apabila admin) dan registrasi (apabila bukan admin), admin akan dapat melakukan 5 hal, sedangkan *user* hanya dapat melakukan 2 hal. Admin dapat ; Menambahkan atlet kedalam daftar manajemen, Melihat daftar manajemen, Mengedit daftar manajemen, Menghapus, dan Keluar dari menu program, sedangkan *user* hanya dapat melihat dan Keluar dari program.

Saya sebagai Praktikan membuat sekaligus menyelesaikan program ini menggunakan *draw.io* untuk membuat *Flowchart* dan *Visual Studio Code* untuk membuat **kodingannya**.

Flowchart



Gambar 1 Flowchart

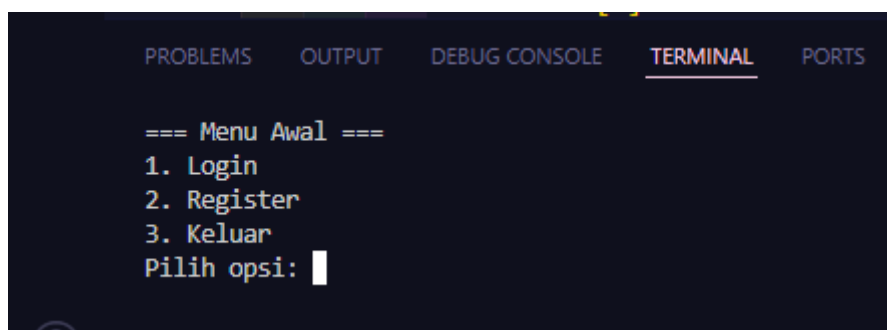
Flowchart ini hanya sebagai Gambaran atau sketsa dari codingan program Manajemen Atlet dan tidak dapat dijalankan atau di-*run*.

Codingan program di Python

```
1 # Fungsi untuk menampilkan menu awal (Login/register)
2 def menu_awal():
3     while True:
4         clear_screen()
5         print("=== Menu Awal ===")
6         print("1. Login")
7         print("2. Register")
8         print("3. Keluar")
9
10    try:
11        pilihan = int(input("Pilih opsi: "))
12        if pilihan == 1:
13            role = login()
14            if role:
15                menu_utama(role)
16        elif pilihan == 2:
17            register()
18        elif pilihan == 3:
19            clear_screen()
20            print("Terima kasih telah menggunakan aplikasi!")
21            break
22        else:
23            print("\nPilihan tidak valid.")
24    except ValueError:
25        print("\nInput tidak valid. Harap masukkan angka.")
26    input("\nTekan Enter untuk melanjutkan...")
```

```
1 # List untuk menyimpan data user (username, password, role)
2 users = [["admin", "admin123", "admin"]]
3 atlet_list = []
4
5 # Fungsi untuk Login
6 def login():
7     clear_screen()
8     print("--- Login ---")
9     username = input("Username: ")
10    password = input("Password: ")
11
12    # Cek user di dalam database users
13    for user in users:
14        if user[0] == username and user[1] == password:
15            print(f"\nSelamat datang, {username}!")
16            return user[2]
17
18    print("\nUsername atau password salah!")
19    input("Tekan Enter untuk kembali ke login...")
20    return None
21
```

Gambar 2.1



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

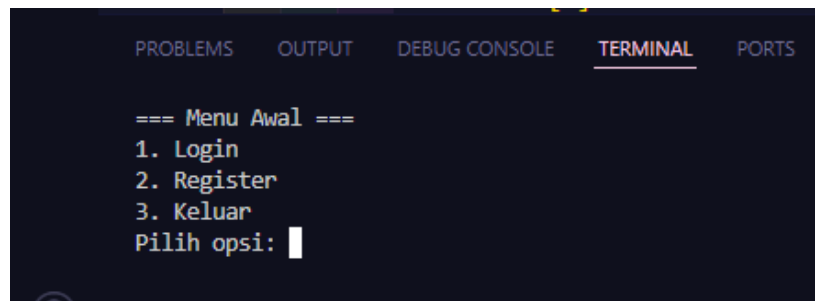
=== Menu Awal ===
1. Login
2. Register
3. Keluar
Pilih opsi: |
```

Pada bagian ini pengguna akan melakukan login dengan username: admin,
password: admin123

A screenshot of a code editor with a dark background and light-colored text. The code is a Python function named 'register' that handles user registration. It includes comments in Indonesian, a loop to check if a username already exists, and a list to store user data. The function uses 'input' for user input and 'print' for output. The code is numbered from 1 to 17.

```
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[0] == username:
11             print("\nUsername sudah ada!")
12             input("Tekan Enter untuk kembali ke menu...")
13             return
14
15     users.append([username, password, "user"])
16     print(f"\nPengguna {username} berhasil didaftarkan!")
17     input("Tekan Enter untuk kembali ke menu...")
```

Gambar 2.2

A screenshot of a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected), and 'PORTS'. The terminal content shows a menu titled '=== Menu Awal ===' with three options: '1. Login', '2. Register', and '3. Keluar'. Below the options, it says 'Pilih opsi:' followed by a cursor.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

=== Menu Awal ===
1. Login
2. Register
3. Keluar
Pilih opsi: |
```

Pengguna akan melakukan registrasi apabila pengguna bukan admin

```

1  # Fungsi utama untuk menampilkan menu
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 = int(input("Pilih opsi: "))
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 ValueError:
30             print("\nInput tidak valid. Harap masukkan angka.")
31             input("\nTekan Enter untuk melanjutkan...")

```

Gambar 2.3

```

PROBLEMS  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: █

```

Pengguna akan diminta untuk memilih antara 5 pilihan apabila pengguna adalah admin

```
PROBLEMS OUTPUT DEBUG CONSOLE

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

Pengguna akan diminta untuk memilih antara 2 pilihan apabila pengguna adalah user

```
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 = int(input("Masukkan Usia Atlet: "))
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 list
14        atlet = [nama, usia, cabang, posisi, statistik, prestasi]
15
16        # Menambahkan atlet ke dalam list
17        atlet_list.append(atlet)
18        print(f"\nAtlet {nama} berhasil ditambahkan!\n")
19    except ValueError:
20        print("\nInput usia harus berupa angka!")
21
22    input("Tekan Enter untuk kembali ke menu...")
```

Gambar 2.4

Pengguna akan diminta untuk memasukkan identitas atlet, seperti nama, usia, cabang, posisi, statistik, dan prestasi

```
1 # Fungsi untuk melihat daftar atlet (nama - usia - cabang - 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"{index + 1}. {atlet[0]} - {atlet[1]} - {atlet[2]} - {atlet[3]} - {atlet[4]} - {atlet[5]}")
10        input("\nTekan Enter untuk kembali ke menu...")
```

Gambar 2.5

Pengguna akan diperlihatkan daftar atlet yang telah ditambahkan

```
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 = int(input("Pilih nomor atlet yang ingin diedit: "))
8             if 1 <= pilihan <= Len(atlet_list):
9                 atlet = atlet_list[pilihan - 1]
10                print(f"\n--- Edit Informasi Atlet {atlet[0]} ---")
11                atlet[0] = input(f>Nama ({atlet[0]}): ") or atlet[0]
12                atlet[1] = input(f"Usia ({atlet[1]}): ") or atlet[1]
13                atlet[2] = input(f"Cabang ({atlet[2]}): ") or atlet[2]
14                atlet[3] = input(f"Posisi ({atlet[3]}): ") or atlet[3]
15                atlet[4] = input(f"Statistik ({atlet[4]}): ") or atlet[4]
16                atlet[5] = input(f"Prestasi ({atlet[5]}): ") or atlet[5]
17                print(f"\nAtlet {atlet[0]} 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...")
```

Gambar 2.6

Pengguna akan mengedit identitas dari atlet entah itu nama, usia, cabor, dan semacamnya

```
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 = int(input("Pilih nomor atlet yang ingin dihapus: "))
9             if 1 <= pilihan <= Len(atlet_list):
10                atlet = atlet_list.pop(pilihan - 1)
11                print(f"\nAtlet {atlet[0]} 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 2.7

Pengguna akan menghapus atlet dari daftar manajemen

Sekian dari saya

Terima kasih

Maaf terlalu singkat bang