

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN DASAR
POSTTEST 6



Informatika B'24
Rahmat Riyadi
2409106074

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2024

PEMBAHASAN

1.1 LATAR BELAKANG

1.1.1 Pendahuluan

Menulis kode yang sama secara berulang akan membuang-buang waktu dan bakal memperlambat python dalam mengeksekusi kode. Dalam mengatasi hal ini terdapat suatu hal yang dapat menyimpan kode tersebut dan memungkinkan kita memanggil untuk menggunakannya kapan saja. Setidaknya itulah definisi sederhana sebuah Fungsi yang memiliki kegunaan agar seorang programmer dapat menggunakan program berkali-kali.

Fungsi didefinisikan sebagai unit modular yang dirancang untuk menjalankan tugas spesifik dalam bahasa pemrograman. Sebenarnya fungsi ini sangat dekat dengan kita dan bahkan saat pertama kali belajar menuliskan “hello world” di python, kita telah menggunakan fungsi yakni `print()`. Fungsi walau tidak dibuat secara juga dapat dipanggil di mana saja contohnya `print()`, `len()` dan sebagainya apabila ditelusuri menggunakan F12. Jika ingin membuat sebuah fungsi cukup didefinisikan menggunakan kata kunci `def` diikuti dengan nama fungsi, kemudian tanda kurung buka dan tutup serta titik dua [`def ini_fungsi():`].

Prosedur mirip dengan fungsi secara sekilas, unit ini merujuk pada serangkaian instruksi yang dieksekusi untuk mencapai suatu tugas, tanpa mengembalikan nilai. Prosedur umumnya digunakan untuk mendefinisikan suatu pernyataan mandiri, atau bisa dibilang hanya proses-proses sederhana. Operasi yang biasanya dilakukan prosedur seperti mencetak data atau memperbarui nilai variabel. Sebagai salah satu unit yang juga menggunakan `def`, prosedur dapat dipanggil di mana saja. Sama dengan fungsi, jika kita melakukan perubahan pada `def`, itu akan mempengaruhi seluruhnya.

Kita telah mengetahui bahwa sesuatu yang didefinisikan dengan unit `def` itu bisa berupa fungsi ataupun prosedur. Keduanya memang cukup sulit dibedakan, namun yang jelas fungsi pasti memiliki `return`, sedangkan prosedur adalah blok yang tidak mengembalikan nilai. Tetapi berdasarkan informasi dari *dokumentasi resmi python* baik fungsi yang mengembalikan nilai atau tidak, semuanya disebut fungsi. Dalam bahasa lain keduanya mungkin berbeda, namun dalam kesederhanaan python keduanya adalah satu kesatuan yang didefinisikan menggunakan unit `def` (define).

Variabel lokal dan global adalah dijumpai pada fungsi dan prosedur ini. Variabel lokal merupakan suatu variabel yang dideklarasikan di luar,

sehingga tersedia untuk digunakan semua fungsi. Sedangkan variabel lokal dideklarasikan di dalam sebuah fungsi dan hanya dapat diakses oleh fungsi itu sendiri.

1.1.2 Masalah

Posttest 7 adalah perpanjangan dari posttest 6, setelah mempelajari mengenai fungsi, prosedur, variabel luar dan dalam, di sinilah hal tersebut diterapkan. Pada tugas diminta untuk membuat 3 fungsi dan 2 prosedur, menggunakan minimal 3 variabel global dan 5 variabel lokal. Poin plus akan diberikan apabila dapat menggunakan fungsi rekursif dan error handling pada program.

1.1.3 Pemecahan Masalah

Masalah utama saya awalnya adalah membedakan fungsi dan prosedur, hal tersebut membuat saya mencoba belajar kembali dan mencoba memahaminya dari beberapa literatur di internet. Keduanya memang menggunakan unit yang sama yakni `def`, sehingga hal tersebut hampir saja membuat kesalahan yang fatal. Selanjutnya juga mengenai fungsi rekursif yang juga perlu dipelajari lebih lanjut sebelum diterapkan dalam program.

Masalah selanjutnya yakni dalam penerapan fungsi dan prosedur dalam program. Awalnya saya hanya berpikir untuk membuat fungsi untuk program yang dapat diakses oleh admin dan oleh user. Namun untuk memenuhi instruksi tugas, maka banyak dari fitur telah dibuatkan fungsinya sendiri-sendiri, terutama yang kelihatan cukup ribet.

Kemudian dari sini muncul masalah baru yang baru aku sadari pada input pilihan dalam programku. Di mana input huruf pada program yang di inisialisasi sebagai integer akan menghasilkan error apabila di inputkan huruf. Sehingga solusi yang harus diambil adalah dengan menggunakan `try` dan `except` yang sebelumnya telah aku pelajari dari IDCAMP.

Ternyata ada masalah lanjutan akibat mengikuti rekursif. Fitur yang memungkinkan pengguna untuk kembali ke menu pilihan awal menjadi hilang, sehingga apabila pengguna memilih login sebagai admin ia akan terus login sebagai admin hingga kesempatannya habis. Memang tidak sempat mencari solusi untuk ini, sehingga dibiarkan saja, apalagi di bagian sini jugalah terpaksa menggunakan fungsi `exit()`. Tetapi tidak apa solusi yang memberikan masalah baru, yang penting tugas terselesaikan dan dapat melanjutkan tugas yang baru.

Masih ada satu hal, untuk memenuhi syarat fungsi yang minimal 3 aku menciptakan sebuah fitur baru. Demi menyelesaikan masalah ini, fitur baru yang memungkinkan user melihat jumlah anime yang tayang pada musim tersebut. Walau kurang berpengaruh untuk jumlah anime yang hanya 10, tapi fitur ini pasti akan sangat berpengaruh jika jumlah animenya cukup banyak, jadi pasti akan dibutuhkan.

Ternyata masih ada masalah lain tetapi kali ini pada bagian flowchart. Fungsi dan prosedur ternyata memiliki simbol khusus untuk mendefinisikannya. Namun dari pengamatan dan hasil diskusi dengan teman-teman, mereka juga kebingungan dan banyak juga yang menggunakan flowchart sebelumnya. Jadi pada masalah kali ini, solusi yang aku ambil adalah dengan mengikuti jalan tercepat agar dapat mengerjakan tugas lain dan mengikuti bootcamp.

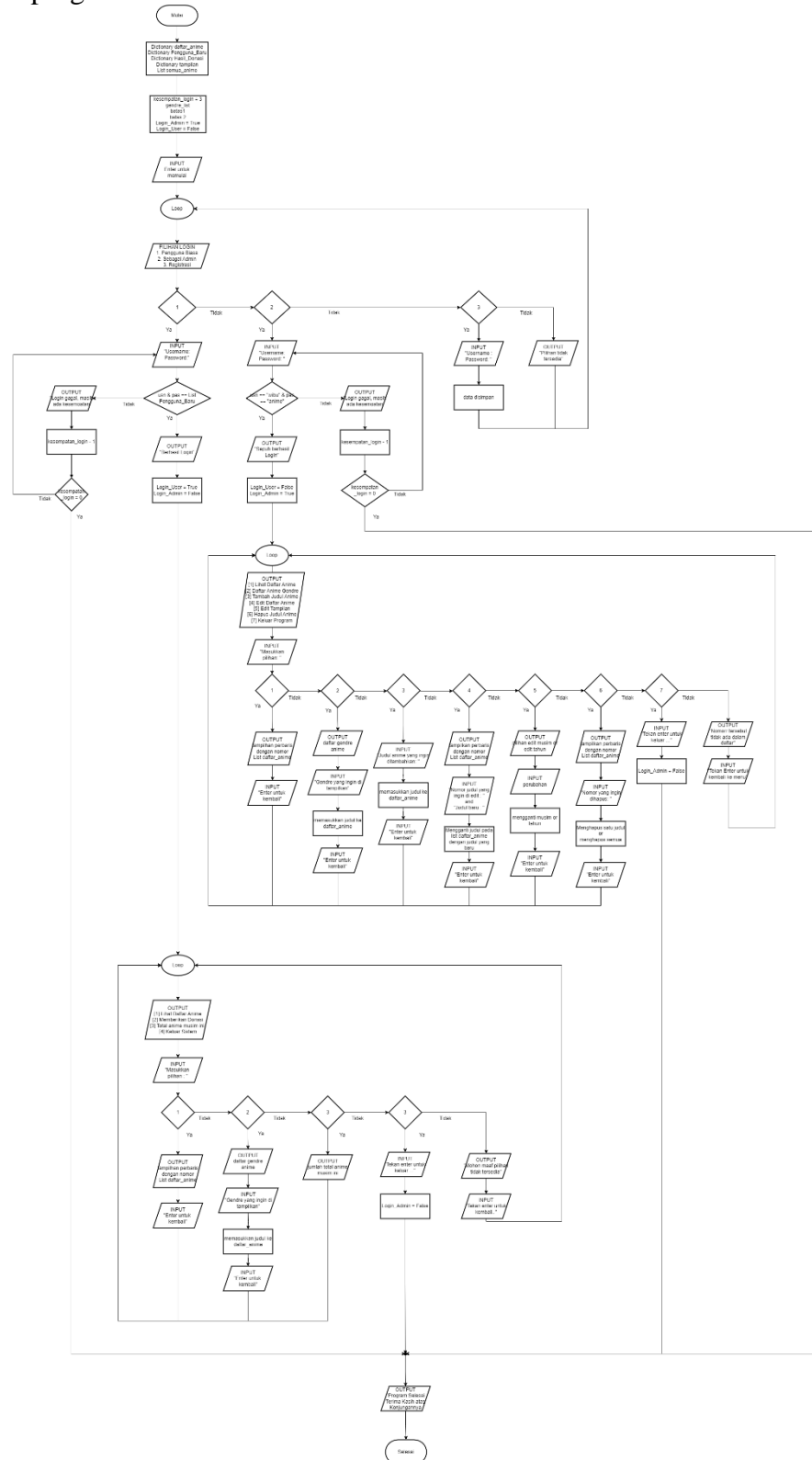
Terakhir untuk login sebagai Admin, masih sama tetap menggunakan username dan password khusus. Sebenarnya akan lebih baik jika admin juga dapat melakukan login, hal ini akan diterapkan lain waktu. Saat ini pada program admin memiliki username *wibu* dengan password *anime*.

LOGIN ADMIN

Username	wibu
Password	anime

1.2 FLOWCHART

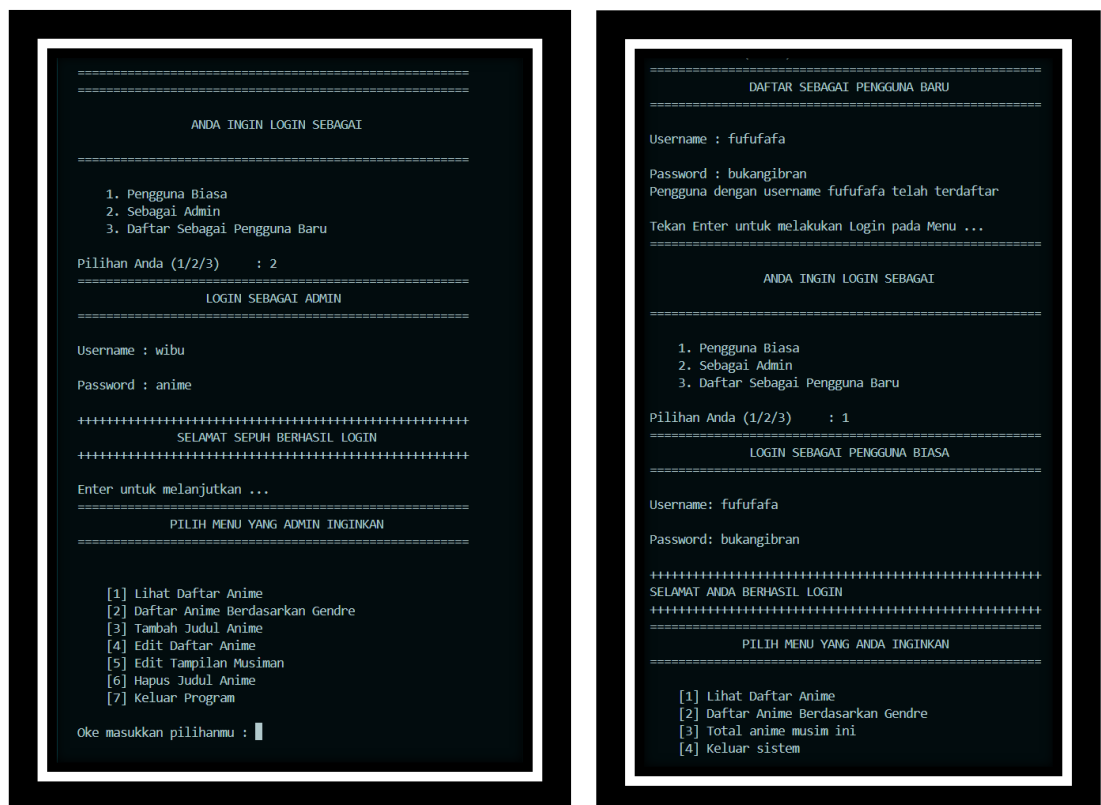
Berikut gambar flowchart posttest sebelumnya yang telah dimodifikasi berdasarkan program saat ini.



1.3 OUTPUT PROGRAM

Tampilan terminal kali ini juga mirip dengan laporan sebelumnya, jadi yang akan ditampilkan di sini hanyalah sesuatu yang berbeda dan unik dibandingkan sebelumnya.

1.3.1 Login Admin dan User



```
=====
=====
                          ANDA INGIN LOGIN SEBAGAI
=====

1. Pengguna Biasa
2. Sebagai Admin
3. Daftar Sebagai Pengguna Baru

Pilihan Anda (1/2/3)      : 2
=====
                          LOGIN SEBAGAI ADMIN
=====

Username : wibu
Password : anime

+++++
SELAMAT SEPUH BERHASIL LOGIN
+++++

Enter untuk melanjutkan ...

=====
                          PILIH MENU YANG ADMIN INGINKAN
=====

[1] Lihat Daftar Anime
[2] Daftar Anime Berdasarkan Gendre
[3] Tambah Judul Anime
[4] Edit Daftar Anime
[5] Edit Tampilan Musiman
[6] Hapus Judul Anime
[7] Keluar Program

Oke masukkan pilihanmu : █

=====
=====
                          DAFTAR SEBAGAI PENGGUNA BARU
=====

Username : fufufafa
Password : bukangibran
Pengguna dengan username fufufafa telah terdaftar

Tekan Enter untuk melakukan Login pada Menu ...

=====
                          ANDA INGIN LOGIN SEBAGAI
=====

1. Pengguna Biasa
2. Sebagai Admin
3. Daftar Sebagai Pengguna Baru

Pilihan Anda (1/2/3)      : 1
=====
                          LOGIN SEBAGAI PENGGUNA BIASA
=====

Username: fufufafa
Password: bukangibran

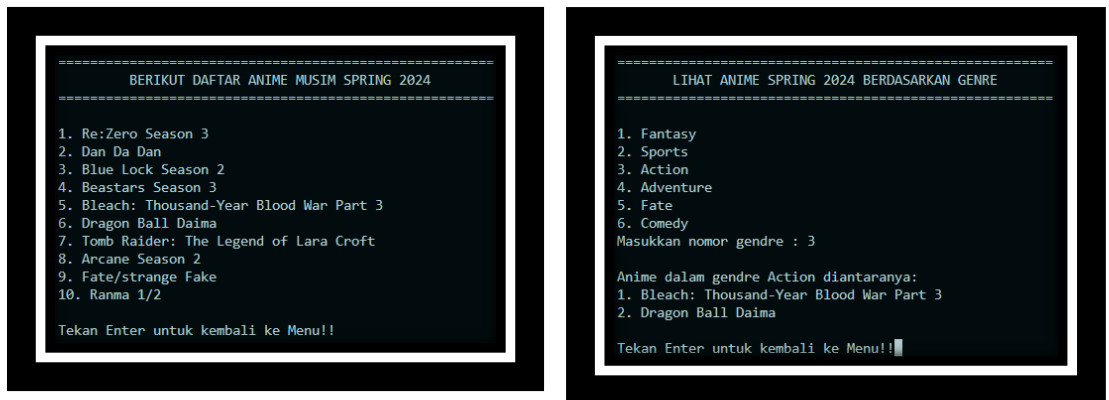
+++++
SELAMAT ANDA BERHASIL LOGIN
+++++

=====
                          PILIH MENU YANG ANDA INGINKAN
=====

[1] Lihat Daftar Anime
[2] Daftar Anime Berdasarkan Gendre
[3] Total anime musim ini
[4] Keluar sistem
```

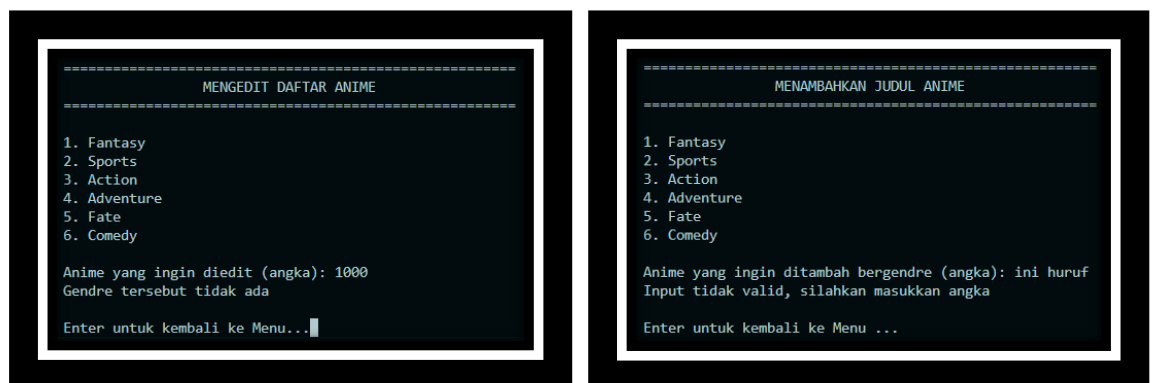
Pada menu user utama user setelah login kali ini bertambah. Fitur menampilkan total anime ini ditambahkan untuk melengkapi persyaratan minimal 3 prosedur. Sebetulnya fitur ini sangat sederhana, menggunakan perulangan for sederhana dan intinya hanya untuk mendapatkan return.

1.3.2 Tampilan Musim dan Tahun Semakin Diperluas



Lebih ditujukan kepada user namun juga dapat diakses oleh admin untuk melihat perubahan yang dilakukan. Tampilan nama musim dan tahun ini penting sekali diletakkan di sini agar dapat menekankan bahwa anime tersebut tayangnya di musim apa dan tahun berapa.

1.3.3 Optimalisasi Error Handling



Kali ini error handling juga mendapatkan perhatian. Pada program sebelumnya program yang dipaksa input int dapat dibatasi sehingga angka yang tidak masuk dalam batas akan termuat dalam else. Namun, apabila memasukkan huruf ada langsung enter maka akan terjadi error, maka digunakanlah try dan excape. Sederhananya try akan mencoba program, dan excape akan memberikan respon error sehingga program tidak terhenti begitu saja.