

LAPORAN PRAKTIKUM POSTTEST 7
ALGORITMA DAN PEMROGRAMAN DASAR



Dimas Firjatullah Islamay
B1

2409106057

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

2024

LATAR BELAKANG

Pada soal posttest ketujuh ini meminta saya membuat flowchart dan program Python yang berhubungan dengan fungsi. saya disuruh membuat program yang berhubungan dengan fungsi yang terdiri dari fungsi dengan atau tanpa parameter, menggunakan prosedur, dan membuat variabel global dan lokal dengan tema yang sama dengan posttest 5 dan 6 yaitu manajemen tanaman hias.

Saya akan membuat flowchart menggunakan draw.io dan program dengan menggunakan Python. Saya juga menyelipkan beberapa percabangan, list, sedikit perulangan, dan lain lain ke dalam program Python saya.

SOLUSI

Pada bagian ini saya akan menjawab serta menjelaskan sebisa saya bagaimana saya menjawab soal yang telah diberikan, langkah - langkahnya bagaimana, dll. Mulai dari flowchart terlebih dahulu lalu ke pemrograman Python. Berikut soal Post-Test 7 :

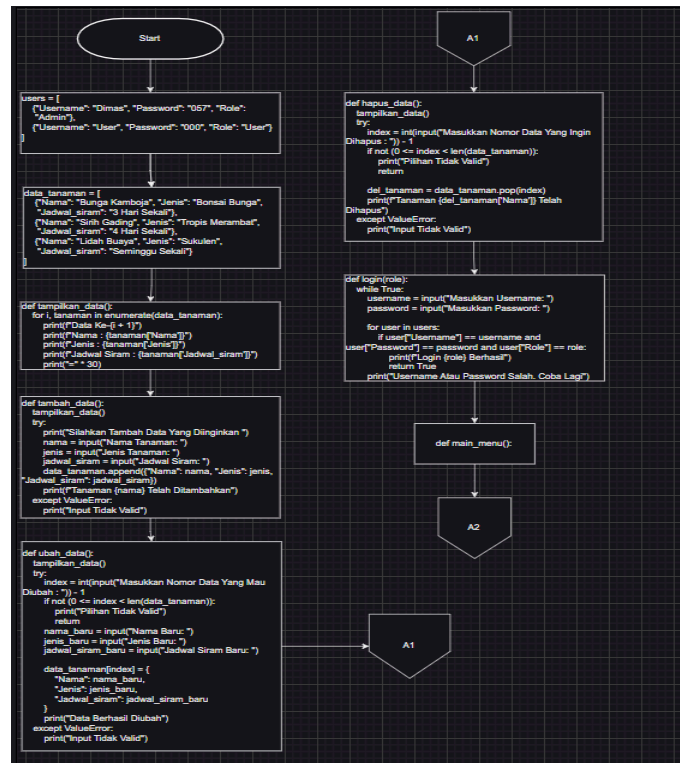
INSTRUKSI:

Lanjutkan program yang telah kalian buat pada Posttest 6, dengan ketentuan sebagai berikut:

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

Buat juga flowchart dari program yang kalian buat.

FLOWCHART



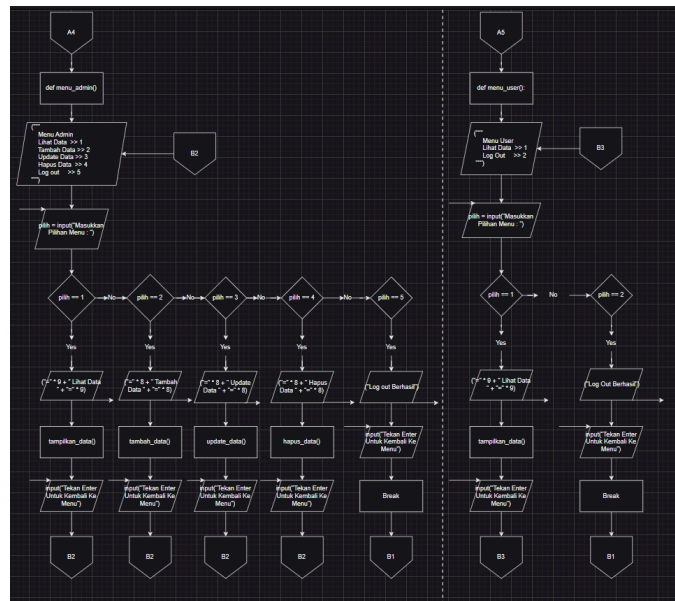
Gambar 1

1. Langkah pertama saya akan buat dulu list yang berisi users dan data_tanaman dan beberapa fungsi def agar nanti bisa saya panggil kembali, di antaranya adalah fungsi def tampilkan_data, tambah_data, update data, hapus_data, login(role), dan main_menu yang isinya akan saya jelaskan masing-masing di langkah-langkah selanjutnya.
2. Di dalam list users berisi 2 jenis user yaitu admin dengan nama Dimas dan password 057 (yang dapat menampilkan, menambah, mengubah, dan menghapus data), serta user dengan nama User dan password 000 (yang hanya dapat menampilkan data).
3. Di dalam list data_tanaman terdapat 3 nama tanaman, jenis tanaman, dan jadwal_siram tanaman sesuai dengan posttest sebelumnya.

4. Di dalam fungsi `def tampilkan_data` terdapat `enumerate`. `Enumerate` ini berfungsi untuk mengiterasi dan memberi nomor urut untuk masing-masing data. Kemudian saya buat `print(f'Data Ke-{i + 1}')`. Maksudnya adalah dia akan mencetak nomor data yang dimulai dari 1 bukan 0, setelah itu disesuaikan saja dengan menggunakan `f` sebelum tanda “ biar bisa memasukkan data dari `data_tanaman` yang ada di list paling awal, contohnya seperti `f' Nama` untuk nama tanaman, `f' Jenis` untuk jenis tanaman, `f' Jadwal Siram` untuk `jadwal_siram` tanaman.
5. Di dalam `def tambah_data` terdapat `data_tanaman_append`. `append` disini berfungsi untuk menambahkan item baru ke dalam list. jadi yang pertama saya buat input dulu untuk menambahkan nama, jenis dan `jadwal_siram` tanaman baru, lalu saya berikan `data_tanaman_append` dengan format `({"Nama": nama, "Jenis": jenis, "Jadwal_siram": jadwal_siram})` artinya kita akan menambahkan nama baru di bagian Nama dalam `data_tanaman`, jenis baru di bagian Jenis dalam `data_tanaman` dan `jadwal_siram` baru di bagian `Jadwal_siram` dalam `data_tanaman`. Saya juga menambahkan fungsi `try` dan `except` untuk memperbaiki kesalahan yang terjadi.
6. Di dalam `def update_data` terdapat `tampilkan_data()`, `index`, dan `len(data_tanaman)`. `tampilkan_data()` ini berfungsi untuk memanggil `def tampilkan_data` di langkah keempat tadi agar bisa masuk kesini, tujuannya adalah karena saya mau sebelum admin dapat mengupdate data, dia bisa melihat dulu menu `data_tanamannya` agar dia tau yang mau di update yang mana, makanya saya memasukkan kembali `def tampilkan_data` disini. `Index` disini berfungsi untuk memilih nomor untuk data yang mau diubah saya beri `-1` di ujung kalimat karena `index` mulainya dari 0, setelah itu terdapat `if not (0 <= index < len(data_tanaman))`: maksudnya adalah bahwa `index` harus lebih besar sama dengan 0 dan lebih kecil dari `len`, karena `len` sendiri merupakan jumlah total elemen dalam `data_tanaman`. kurang lebih `len` ini sebagai patokannya biar `indexnya` gak keluar batas. Kemudian saya buat input nama, jenis, dan `jadwal_siram` tanaman sebagai data terbaru yang akan dimasukkan, lalu saya buat simbol proses

`data_tanaman[index] = { "Nama": nama_baru, "Jenis" :jenis_baru, "Jadwal_siram": jadwal_siram_baru}` untuk memperbarui data dari `data_tanaman` dan agar tidak keluar batas. Saya juga menambahkan fungsi `try` dan `except` untuk memperbaiki kesalahan yang terjadi.

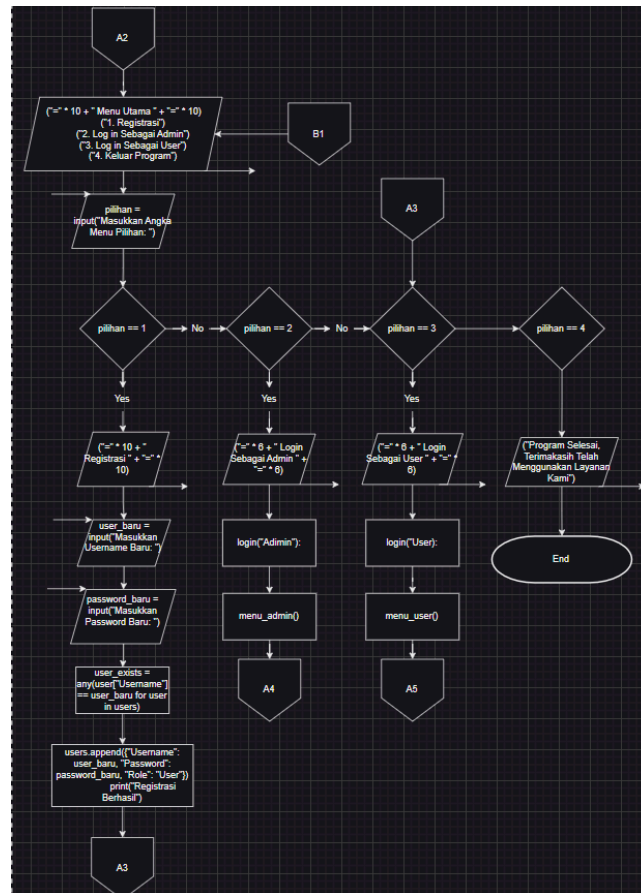
7. Di dalam `def hapus_data` terdapat `del_tanaman` dan `data_tanaman.pop(index)` yang berfungsi untuk menghapus elemen yang diinginkan dari `data_tanaman` berdasarkan posisinya yang sudah di index tadi. Pertama saya buat input data yang mau dihapus, diteruskan dengan fungsi `if not (0 <= index < len(data_tanaman)):` dan `del_tanaman = data_tanaman.pop(index)` agar dapat menghapus data yang diinginkan. Saya juga menambahkan fungsi `try` dan `except` untuk memperbaiki kesalahan yang terjadi.
8. Di dalam `def login(role)` saya buat parameter `role` agar sistem dapat menentukan apakah ini termasuk Admin atau User, contoh kita ingin masuk sebagai Admin maka `login(role)` ini akan berubah menjadi `login(Admin)` jika kita benar memasukkan username dan passwordnya. lalu saya buat input untuk memasukkan username dan password, setelah itu saya buat `for user in users: if user["Username"] == username and user["Password"] == password and user["Role"] == role: print(f"Login {role} Berhasil") return True`. Ini artinya selama username dan password yang dimasukkan sesuai maka akan menghasilkan hasil `True` dan bisa lanjut ke langkah selanjutnya.



Gambar 2

9. Kemudian terdapat def menu_admin() yang berisi menu admin yang terdiri dari, 1. Lihat Data, 2. Tambah Data, 3. Update Data, 4. Hapus Data, dan 5. Log Out.
10. Jika memilih nomor 1 maka kita akan melihat data saja, berarti saya panggil kembali def tampilkan_data diawal tadi untuk diproses, lalu saya berikan input “Tekan Enter Untuk Kembali Ke Menu” dan akan kembali ke menu admin.
11. Jika memilih nomor 2 maka kita akan menambah data, berarti saya panggil kembali def tambah_data diawal tadi untuk diproses, lalu saya berikan input “Tekan Enter Untuk Kembali Ke Menu” dan akan kembali ke menu admin.
12. Jika memilih nomor 3 maka kita akan mengupdate data, berarti saya panggil kembali def update_data diawal tadi untuk diproses, lalu saya berikan input “Tekan Enter Untuk Kembali Ke Menu” dan akan kembali ke menu admin.

13. Jika memilih nomor 4 maka kita akan menghapus data, berarti saya panggil kembali def hapus_data diawal tadi, lalu saya berikan input “Tekan Enter Untuk Kembali Ke Menu” dan akan kembali ke menu admin.
14. Jika memilih nomor 5 maka kita Log Out dari role kita sebagai Admin, lalu saya berikan input “Tekan Enter Untuk Kembali Ke Menu” dan akan kembali ke menu utama, saya tambahkan break agar tidak terjadi perulangan karena disini kita sudah log out.
15. Setelah itu, sebelahnya def menu_admin, terdapat def menu_user() yang berisi menu user yang terdiri dari 1. Lihat Data, dan 2. Log Out.
16. Jika memilih nomor 1 maka kita akan melihat data saja, berarti saya panggil kembali def tampilkan_data diawal tadi untuk diproses, lalu saya berikan input “Tekan Enter Untuk Kembali Ke Menu” dan akan kembali ke menu user.
17. Jika memilih nomor 2 maka kita Log Out, lalu saya berikan input “Tekan Enter Untuk Kembali Ke Menu” dan akan kembali ke menu utama, saya tambahkan break agar tidak terjadi perulangan karena disini kita sudah log out.



Gambar 3

18. Di dalam def main_menu saya buat menu utama yang terdiri dari 1. Registrasi, 2. Log in sebagai Admin, 3. Log in sebagai User, dan 4. Keluar Program. def main menu ini harus ada agar semua codingan ini bisa dijalankan.
19. Setelah itu saya buat input, dengan variabel pilihan dan saya buat simbol decision yang terdiri dari nomor 1 sampai 4.
20. Jika kita memilih nomor 1 artinya kita registrasi yang dimana saya buat 2 input yaitu user_baru dan password_baru lalu saya menambahkan `user_exists = any(user["Username"] == user_baru for user in users)` untuk memeriksa apakah pengguna baru memiliki username atau password yang sama dengan username dan password dari pengguna lama, lalu saya

menggunakan lagi fungsi append untuk menambahkan user, password, dan role dari pengguna yang telah registrasi tadi, yang dimana akan lanjut ke nomor 3 atau langkah 22.

21. jika memilih nomor 2 maka akan log in sebagai Admin, saya tinggal memanggil fungsi def login("Admin"): dan menu_admin(), yang dimana menu admin ini akan melanjutkan ke langkah 9 tadi.
22. jika memilih nomor 3 maka akan log in sebagai User, saya tinggal memanggil fungsi def login("User"): dan menu_user(), yang dimana menu user ini akan melanjutkan ke langkah 15 tadi.
23. Jika memilih nomor 4 maka kita akan benar benar keluar program dan terdapat output ("Program Selesai, Terimakasih Telah Menggunakan Layanan Kami") yang menandakan bahwa kita telah keluar dan selesai.

PYTHON

```
1 users = [  
2     {"Username": "Dimas", "Password": "057", "Role": "Admin"},  
3     {"Username": "User", "Password": "000", "Role": "User"}  
4 ]  
5  
6 data_tanaman = [  
7     {"Nama": "Bunga Kamboja", "Jenis": "Bonsai Bunga", "Jadwal_siram": "3 Hari Sekali"},  
8     {"Nama": "Sirih Gading", "Jenis": "Tropis Merambat", "Jadwal_siram": "4 Hari Sekali"},  
9     {"Nama": "Lidah Buaya", "Jenis": "Sukulen", "Jadwal_siram": "Seminggu Sekali"}  
10 ]  
11  
12  
13 def tampilkan_data():  
14     for i, tanaman in enumerate(data_tanaman):  
15         print(f>Data Ke-{i + 1}")  
16         print(f>Nama : {tanaman['Nama']})  
17         print(f>Jenis : {tanaman['Jenis']})  
18         print(f>Jadwal Siram : {tanaman['Jadwal_siram']})  
19         print("=" * 30)  
20  
21  
22 def tambah_data():  
23     tampilkan_data()  
24     try:  
25         print("Silahkan Tambah Data Yang Diinginkan ")  
26         nama = input("Nama Tanaman: ")  
27         jenis = input("Jenis Tanaman: ")  
28         jadwal_siram = input("Jadwal Siram: ")  
29         data_tanaman.append({"Nama": nama, "Jenis": jenis, "Jadwal_siram": jadwal_siram})
```

Gambar 4

1. Langkah pertama saya buat list users yang berisi username Dimas dengan password : 057 role Admin dan username User dengan password 000 dan role User. Setelah itu saya buat list data_tanaman nya yang berisi nama, jenis dan jadwal siram dari 3 tanaman yang sudah saya pilih.
2. Lalu saya buat fungsi def tampilkan_data yang di dalamnya terdapat enumerate. Enumerate ini berfungsi untuk mengiterasi dan memberi nomor urut untuk masing masing data. Kemudian saya buat print(f>Data Ke-{i + 1}"). Maksudnya adalah dia akan mencetak nomor data yang dimulai dari 1 bukan 0, setelah itu disesuaikan saja dengan menggunakan f sebelum tanda “ biar bisa memasukkan data dari data_tanaman yang ada di list paling awal, contohnya seperti f” Nama untuk nama tanaman, f” Jenis untuk jenis tanaman, f” Jadwal Siram untuk jadwal_siram tanaman.

3. Kemudian saya buat fungsi def tambah_data yang di dalamnya terdapat data_tanaman_append. append disini berfungsi untuk menambahkan item baru ke dalam list. jadi yang pertama saya buat input dulu untuk menambahkan nama, jenis dan jadwal_siram tanaman baru, lalu saya berikan data_tanaman_append dengan format ({"Nama": nama, "Jenis": jenis, "Jadwal_siram": jadwal_siram}) artinya kita akan menambahkan nama baru di bagian Nama dalam data_tanaman, jenis baru di bagian Jenis dalam data_tanaman dan jadwal_siram baru di bagian Jadwal_siram dalam data_tanaman. Saya juga menambahkan fungsi try dan except untuk memperbaiki kesalahan yang terjadi.

```
34
35 def update_data():
36     tampilkan_data()
37     try:
38         index = int(input("Masukkan Nomor Data Yang Mau Diupdate : ")) - 1
39         if not (0 <= index < len(data_tanaman)):
40             print("Pilihan Tidak Valid")
41             return
42         nama_baru = input("Nama Baru: ")
43         jenis_baru = input("Jenis Baru: ")
44         jadwal_siram_baru = input("Jadwal Siram Baru: ")
45
46         data_tanaman[index] = {
47             "Nama": nama_baru,
48             "Jenis": jenis_baru,
49             "Jadwal_siram": jadwal_siram_baru
50         }
51         print("Data Berhasil Diubah")
52     except ValueError:
53         print("Input Tidak Valid")
54
55
56 def hapus_data():
57     tampilkan_data()
58     try:
59         index = int(input("Masukkan Nomor Data Yang Ingin Dihapus : ")) - 1
60         if not (0 <= index < len(data_tanaman)):
61             print("Pilihan Tidak Valid")
62             return
```

Gambar 5

4. Setelah itu saya buat def update_data yang di dalamnya terdapat tampilkan_data(), index, dan len(data_tanaman). tampilkan_data() ini berfungsi untuk memanggil def tampilkan_data di langkah keempat tadi agar bisa masuk kesini, tujuannya adalah karena saya mau sebelum admin dapat mengupdate data, dia bisa melihat dulu menu data_tanamannya agar dia tau yang mau di update yang mana, makanya saya memasukkan

kembali def tampilkan_data disini. Lalu saya bikin 3 input baru dengan variabel nama_baru, jenis_baru, dan jadwal_siram_baru. Index disini berfungsi untuk memilih nomor untuk data yang mau diubah saya beri -1 di ujung kalimat karena index mulainya dari 0, setelah itu terdapat if not (0 <= index < len(data_tanaman)): maksudnya adalah bahwa index harus lebih besar dari 0 dan lebih kecil dari len, karena len sendiri merupakan jumlah total elemen dalam data_tanaman. kurang lebih len ini sebagai patokannya biar indexnya gak keluar batas. Kemudian saya buat simbol proses data_tanaman[index] = { "Nama": nama_baru, "Jenis" :jenis_baru, "Jadwal_siram": jadwal_siram_baru} untuk memperbarui data dari data_tanaman dan agar tidak keluar batas. Saya juga menambahkan fungsi try dan except untuk memperbaiki kesalahan yang terjadi.

5. Kemudian saya buat def hapus_data yang di dalamnya terdapat del_tanaman dan data_tanaman.pop(index) yang berfungsi untuk menghapus elemen yang diinginkan dari data_tanaman berdasarkan posisinya yang sudah di index tadi. Pertama saya buat input data yang mau dihapus, diteruskan dengan fungsi if not (0 <= index < len(data_tanaman)): dan del_tanaman = data_tanaman.pop(index) agar dapat menghapus data yang diinginkan. Saya juga menambahkan fungsi try dan except untuk memperbaiki kesalahan yang terjadi.

```

70 def login(role):
71     while True:
72         username = input("Masukkan Username: ")
73         password = input("Masukkan Password: ")
74
75         for user in users:
76             if user["Username"] == username and user["Password"] == password and user["Role"] == role:
77                 print(f"Login {role} Berhasil")
78                 return True
79         print("Username Atau Password Salah. Coba Lagi")
80
81
82 def main_menu():
83     while True:
84         print("=" * 10 + " Menu Utama " + "=" * 10)
85         print("1. Registrasi")
86         print("2. Log in Sebagai Admin")
87         print("3. Log in Sebagai User")
88         print("4. Keluar Program")
89
90         pilihan = input("Masukkan Angka Menu Pilihan: ")
91
92         if pilihan == "1":
93             print("=" * 10 + " Registrasi " + "=" * 10)
94             user_baru = input("Masukkan Username Baru: ")
95             password_baru = input("Masukkan Password Baru: ")
96             user_exists = any(user["Username"] == user_baru for user in users)
97
98             if user_exists:

```

Gambar 6

6. Selanjutnya saya buat def login(role) buat parameter role agar sistem dapat menentukan apakah ini termasuk Admin atau User, contoh kita ingin masuk sebagai Admin maka login(role) ini akan berubah menjadi login(Admin) jika kita benar memasukkan username dan passwordnya. lalu saya buat input untuk memasukkan username dan password, setelah itu saya buat for user in users: if user["Username"] == username and user["Password"] == password and user["Role"] == role: print(f"Login {role} Berhasil") return True. Ini artinya selama username dan password yang dimasukkan sesuai maka akan menghasilkan hasil True dan bisa lanjut ke langkah selanjutnya.
7. Lalu terdapat def main_menu disertai dengan while True yang didalamnya saya buat menu utama yang terdiri dari 1. Registrasi, 2. Log in sebagai Admin, 3. Log in sebagai User, dan 4. Keluar Program. def main menu ini harus ada agar semua codingan ini bisa dijalankan.
8. Setelah itu saya buat input, dengan variabel pilihan, setelah itu saya buat percabangan. Jika kita memilih nomor 1 artinya kita registrasi yang

dimana saya buat 2 input yaitu user_baru dan password_baru lalu saya menambahkan user_exists = any(user["Username"] == user_baru for user in users) untuk memeriksa apakah pengguna baru memiliki username atau password yang sama dengan username dan password dari pengguna lama, lalu saya menggunakan lagi fungsi append untuk menambahkan user, password, dan role dari pengguna yang telah registrasi tadi, yang dimana akan lanjut ke nomor 3 atau langkah 10.

```

93     print("=" * 10 + " Registrasi " + "=" * 10)
94     user_baru = input("Masukkan Username Baru: ")
95     password_baru = input("Masukkan Password Baru: ")
96     user_exists = any(user["Username"] == user_baru for user in users)
97
98     if user_exists:
99         print("Username Sudah Terdaftar")
100     else:
101         users.append({"Username": user_baru, "Password": password_baru, "Role": "User"})
102         print("Registrasi Berhasil")
103
104     elif pilihan == "2":
105         print("=" * 6 + " Login Sebagai Admin " + "=" * 6)
106         if login("Admin"):
107             menu_admin()
108
109     elif pilihan == "3":
110         print("=" * 6 + " Login Sebagai User " + "=" * 6)
111         if login("User"):
112             menu_user()
113
114     elif pilihan == "4":
115         print("Program Selesai, Terimakasih Telah Menggunakan Layanan Kami")
116         break
117     else:
118         print("Pilihan Tidak Valid")
119         input("Tekan Enter Untuk Kembali")

```

Gambar 7

9. jika memilih nomor 2 maka akan log in sebagai Admin, saya tinggal memanggil fungsi def login("Admin"): dan menu_admin().
10. jika memilih nomor 3 maka akan log in sebagai User, saya tinggal memanggil fungsi def login("User"): dan menu_user().
11. Jika memilih nomor 4 maka kita akan benar benar keluar program dan terdapat output ("Program Selesai, Terimakasih Telah Menggunakan Layanan Kami") yang menandakan bahwa kita telah keluar dan selesai.
12. Jika memilih selain 4 itu maka akan muncul output "Pilihan Tidak Valid".

```

122 def menu_admin():
123     while True:
124         print("""
125             Menu Admin
126             Lihat Data >> 1
127             Tambah Data >> 2
128             Update Data >> 3
129             Hapus Data >> 4
130             Log out >> 5
131         """)
132         pilih = input("Masukkan Pilihan Menu : ")
133
134         if pilih == "1":
135             print("=" * 9 + " Lihat Data " + "=" * 9)
136             tampilkan_data()
137             input("Tekan Enter Untuk Kembali Ke Menu")
138         elif pilih == "2":
139             print("=" * 8 + " Tambah Data " + "=" * 8)
140             tambah_data()
141             input("Tekan Enter Untuk Kembali Ke Menu")
142         elif pilih == "3":
143             print("=" * 8 + " Update Data " + "=" * 8)
144             update_data()
145             input("Tekan Enter Untuk Kembali Ke Menu")
146         elif pilih == "4":
147             print("=" * 8 + " Hapus Data " + "=" * 8)
148             hapus_data()
149             input("Tekan Enter Untuk Kembali Ke Menu")
150         elif pilih == "5":

```

Gambar 8

13. Kemudian terdapat def menu_admin() yang berisi menu admin yang terdiri dari, 1. Lihat Data, 2. Tambah Data, 3. Update Data, 4. Hapus Data, dan 5. Log Out.
14. Jika memilih nomor 1 maka kita akan melihat data saja, berarti saya panggil kembali def tampilkan_data diawal tadi untuk diproses, lalu saya berikan input “Tekan Enter Untuk Kembali Ke Menu” dan akan kembali ke menu admin.
15. Jika memilih nomor 2 maka kita akan menambah data, berarti saya panggil kembali def tambah_data diawal tadi untuk diproses, lalu saya berikan input “Tekan Enter Untuk Kembali Ke Menu” dan akan kembali ke menu admin.
16. Jika memilih nomor 3 maka kita akan mengupdate data, berarti saya panggil kembali def update_data diawal tadi untuk diproses, lalu saya berikan input “Tekan Enter Untuk Kembali Ke Menu” dan akan kembali ke menu admin.

17. Jika memilih nomor 4 maka kita akan menghapus data, berarti saya panggil kembali def hapus_data diawal tadi, lalu saya berikan input “Tekan Enter Untuk Kembali Ke Menu” dan akan kembali ke menu admin.
18. Jika memilih nomor 5 maka kita Log Out dari role kita sebagai Admin, lalu saya berikan input “Tekan Enter Untuk Kembali Ke Menu” dan akan kembali ke menu utama, saya tambahkan break agar tidak terjadi perulangan karena disini kita sudah log out.
19. Jika memilih selain 5 itu maka akan muncul output “Menu Tidak Tersedia”.

```
159 def menu_user():
160     while True:
161         print("""
162             Menu User
163             Lihat Data >> 1
164             Log Out >> 2
165         """)
166         pilih = input("Masukkan Pilihan Menu : ")
167
168         if pilih == "1":
169             print("=" * 9 + " Lihat Data " + "=" * 9)
170             tampilkan_data()
171             input("Tekan Enter Untuk Kembali Ke Menu")
172         elif pilih == "2":
173             print("Log Out Berhasil")
174             input("Tekan Enter Untuk Kembali Ke Menu")
175             break
176         else:
177             print(f"Menu {pilih} Tidak Tersedia")
178             input("Tekan Enter Untuk Kembali Ke Menu")
179
180     main_menu()
181
182
```

Gambar 9

20. Setelah itu, terdapat def menu_user() yang berisi menu user yang terdiri dari 1. Lihat Data, dan 2. Log Out.
21. Jika memilih nomor 1 maka kita akan melihat data saja, berarti saya panggil kembali def tampilkan_data diawal tadi untuk diproses, lalu saya berikan input “Tekan Enter Untuk Kembali Ke Menu” dan akan kembali ke menu user.

22. Jika memilih nomor 2 maka kita Log Out, lalu saya berikan input “Tekan Enter Untuk Kembali Ke Menu” dan akan kembali ke menu utama, saya tambahkan break agar tidak terjadi perulangan karena disini kita sudah log out. Terakhir saya tambahkan main_menu() agar semua programnya jalan.