

LAPORAN PRAKTIKUM POSTTEST 7
ALGORITMA DAN PEMROGRAMAN DASAR



Muhammad Nabil
Rahmatullah B'24

2409106046

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

2024

LATAR BELAKANG

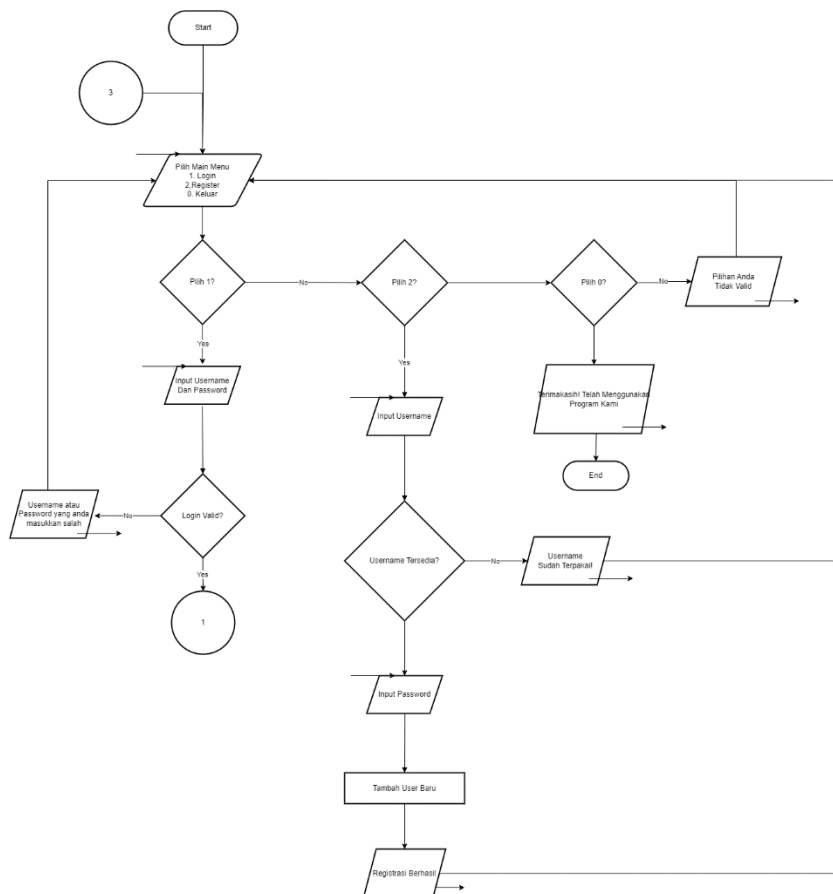
Pada Posttest kali ini penulis diminta untuk membuat program yang bisa melakukan Create, Read, Update, dan Delete (CRUD) dengan ketentuan berikut:

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

Disini saya akan menggunakan tema “Data Penjualan Pada E-commerce” Untuk membuat program CRUD dengan ketentuan tadi.

SOLUSI

A. FLOWCHART

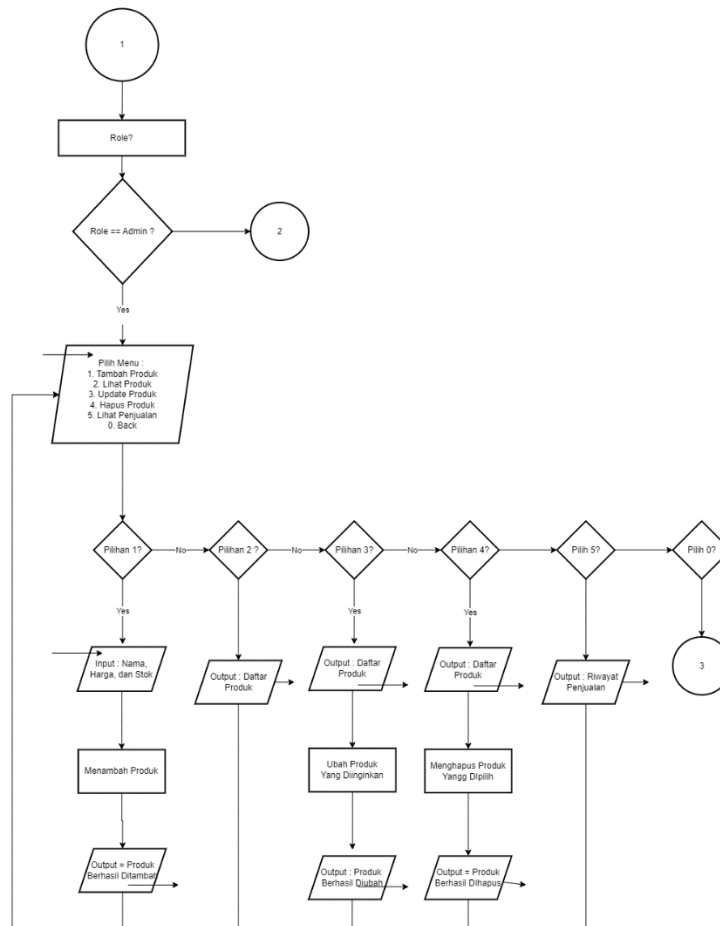


Pada Langkah pertama saya akan memunculkan pilihan yang tersedia di menu sambil meminta *user* untuk melakukan *input* agar program dapat memproses apa yang di mau oleh user.

Kemudian *program* akan meninjau angka yang dipilih user, Jika user melakukan *input* angka 1, maka user diminta untuk memasukkan *Username* dan *Password*, lalu program akan melakukan pengecekan apakah *Username* dan *Password* sudah sesuai dengan yang ada di *Program* tersebut jika iya, program akan melanjutkan ke menu selanjutnya. Namun jika tidak akan kembali ke menu paling pertama.

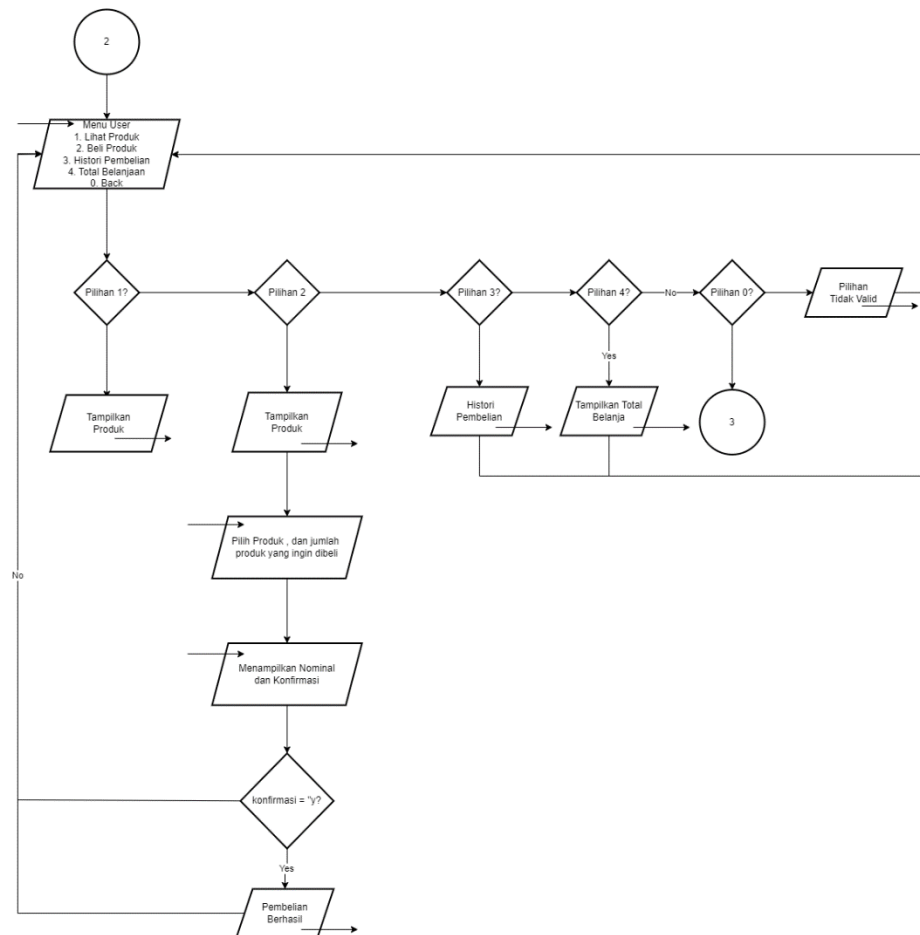
Jika *user* memilih Register maka *user* akan diminta untuk melakukan *input username* dan program akan melakukan pengecekan apakah *username* sudah tersedia di program atau belum, jika sudah maka *user* akan dibawa

kembali ke menu paling pertama. Jika *username* belum terpakai maka *user* akan diminta untuk meng *input* password agar pembuatan akun dapat diproses. Setelah selesai maka *user* akan dibawa kembali ke menu utama dan *user* bisa melakukan login dengan akun yang telah dibuat sebelumnya



Lalu setelah berhasil *login*, *user* akan diperiksa *role* nya terlebih dahulu, jika *role* nya adalah *admin* maka *user* akan dibawa ke menu bagian *admin*.

Pada menu *admin* terdapat 6 pilihan *menu* dan *user* akan diminta melakukan *input* sesuai dengan apa yang *user* pilih. Setelah melakukan hal yang diinginkan oleh *user*, *user* akan dibawa kembali kebagian *menu* yang kedua. Yaitu *menu* yang sekarang.



Lalu jika ketika diperiksa dan *role user* bukan *admin* maka *user* hanya diberi 4 pilihan *menu*. *User* hanya bisa melihat produk dan histori pembelian serta membeli produk yang tersedia di toko. Setelah selesai melakukan pembelian atau apapun yang *user* pilih, *program* akan membawa *user* kembali pada *Menu User* hingga *user* memilih untuk “Back”, dan *user* akan kembali ke *Menu* paling pertama.

B. PYTHON

```
1  users = {
2      "admin": {"password": "admin123", "role": "admin"},
3  }
4
5  produk = {
6      1: {"nama": "Facewash", "harga": 200000, "stok": 50},
7      2: {"nama": "Liptint", "harga": 500000, "stok": 50},
8      3: {"nama": "Serum", "harga": 500000, "stok": 30}
9  }
10
11  listTerjual = []
12
```

Disini saya membagi *file* menjadi 4 agar mempermudah saya dalam pembuatan dan memperbaiki jika terjadi *error*. Pada *file* pertama saya akan membuat *data* menggunakan dictionary yang diperlukan. Disini saya menggunakan *nested dictionary* pada bagian *data*, dikarenakan kita membutuhkan *dictionary* dalam *dictionary* ini untuk kedepannya.

```

1  import os
2  from data import users
3
4  def clear():
5      os.system('cls || clear')
6
7  def login():
8      clear()
9      print("=" * 25, "LOGIN", "=" * 25)
10     username = input("Username: ")
11     password = input("Password: ")
12
13     if username in users and users[username]["password"] == password:
14         return {
15             "username": username,
16             "password": users[username]["password"],
17             "role": users[username]["role"]
18         }
19
20     print("Username atau password salah! Kembali ke menu utama.")
21     input("Tekan Enter untuk melanjutkan...")
22     return None
23
24 def register():
25     try:
26         clear()
27         print("=" * 25, "REGISTER", "=" * 25)
28         username = input("Username baru: ")
29
30         if username in users:
31             print("Username sudah digunakan!")
32             input("Tekan Enter untuk melanjutkan...")
33             return
34
35         password = input("Password: ")
36         users[username] = {"password": password, "role": "user"}
37         print("Sukses!")
38         input("Tekan Enter untuk melanjutkan...")
39     except Exception as e:
40         print(f"Terjadi kesalahan saat registrasi: {str(e)}")
41         input("Tekan Enter untuk melanjutkan...")
42

```

Lalu pada *file* yang kedua saya akan melakukan *autentikasi* atau biasa kita kenal dengan *login*, disini saya melakukan *import os* terlebih dahulu dan saya melakukan “def clear()” agar saya tidak perlu menulis keseluruhan kode untuk melakukan pembersihan pada *terminal*. Dan saya menulis “from data import users” yang berarti saya melakukan *import dictionary users* dari *file data*.

Untuk verifikasi *login* saya menggunakan “if username in users and users[username][“password”] == password” yang berarti *input* yang diberikan *user* harus sama dengan yang ada pada *dictionary* agar *program* dapat lanjut ke menu utamanya

Lalu saya menggunakan “if username in users” untuk melakukan pemeriksaan apakah *username* sudah ada pada *dictionary* atau belum.

```
1 import pandas as pd
2 import time
3 from data import produk, listTerjual
4 from login import clear
5
6 def load(panjangLoading=30, waktu=3.5):
7     for i in range(panjangLoading + 1):
8         time.sleep(waktu / panjangLoading)
9         bar = "=" * i + "-" * (panjangLoading - i)
10        print(f"\rLoading: [{bar}] {i * 100 // panjangLoading}%", end="")
11
12 def showPr():
13     clear()
14     print("-" * 25, "DAFTAR PRODUK", "-" * 25)
15
16     try:
17         if produk:
18             df = pd.DataFrame.from_dict(produk, orient='index', columns=['nama', 'harga', 'stok'])
19             df.index.name = 'No'
20             df.reset_index(inplace=True)
21             print(f"\n{'No':<5} {'Nama':<17} {'Harga (IDR)':<15} {'Stok':<4}")
22             print("=" * 65)
23             for _, row in df.iterrows():
24                 print(f"{'row['No']':<5} {'row['nama']':<17} {'row['harga']':<15,} {'row['stok']':<4}")
25         else:
26             print("Tidak ada produk tersedia.")
27     except Exception as e:
28         print(f"Terjadi kesalahan saat menampilkan produk: {str(e)}")
```

Pada file ke 3 saya akan menuliskan sistem utama dari *program* ini.

Disini saya melakukan *import pandas as pd* dan *import time* untuk memudahkan saya membuat program kedepannya. Saya melakukan “from data import produk, dictionaryTerjual” dan “from login import clear” agar saya tidak perlu menulis ulang bagian *data* dan *clear* lagi.

Lalu saya melakukan *def* pada *codingan* yang panjang agar memudahkan saya jika ingin memanggil ulang kode yang diinginkan. Pada “showPr()” saya menggunakan *df* yaitu fungsi dari *pandas* agar mempermudah saya dalam membuat tabel, dan saya menggunakan “:<17 “ dan sebagainya untuk meratakan teks-teks yang tidak terisi penuh oleh teks dari *dictionary*. Begitu pula pada bagian “noMor” saya menggunakan *df.values* agar mengembalikan pada bagian *DataFrame* yang telah diolah balik menjadi data mentah seperti semula, agar mempermudah saya dalam melakukan *print* secara rinci.


```

1  def addPr():
2      try:
3          clear()
4          print("=" * 9, "TAMBAH PRODUK", "=" * 9)
5          nama = input("Nama produk: ")
6          harga = int(input("Harga produk: "))
7          stok = int(input("Stok produk: "))
8
9          No = max(produk.keys()) + 1 if produk else 1
10         produk[No] = {"nama": nama, "harga": harga, "stok": stok}
11         print("Produk berhasil ditambahkan!")
12         input("Tekan Enter untuk melanjutkan...")
13     except ValueError:
14         print("Harga dan stok harus berupa angka!")
15         input("Tekan Enter untuk melanjutkan...")
16     except Exception as e:
17         print(f"Terjadi kesalahan: {str(e)}")
18         input("Tekan Enter untuk melanjutkan...")
19
20 def updPr():
21     try:
22         showPr()
23         No = int(input("\nMasukkan No produk yang akan diupdate: "))
24
25         if No in produk:
26             nama = input("Nama baru (kosongkan jika tidak ingin diubah): ")
27             if nama:
28                 produk[No]["nama"] = nama
29
30             harga = input("Harga baru (kosongkan jika tidak ingin diubah): ")
31             if harga:
32                 produk[No]["harga"] = int(harga)
33
34             stok = input("Stok baru (kosongkan jika tidak ingin diubah): ")
35             if stok:
36                 produk[No]["stok"] = int(stok)
37
38             print("Produk berhasil diupdate!")
39         else:
40             print("Produk tidak ditemukan!")
41             input("Tekan Enter untuk melanjutkan...")
42     except ValueError:
43         print("No produk, harga, dan stok harus berupa angka!")
44         input("Tekan Enter untuk melanjutkan...")
45     except Exception as e:
46         print(f"Terjadi kesalahan: {str(e)}")
47         input("Tekan Enter untuk melanjutkan...")

```

Kemudian saya melakukan *def* untuk bagian “menambah, mengganti, dan menghapus produk” agar pada saat saya membutuhkan kode tersebut saya hanya perlu menulis huruf pada bagian belakang dari *def*, contohnya saya ingin membuat program di bagian tambah saya hanya perlu menulis “*addPr()*” dan semua kode yg berada di blok “*def addPr()*” akan tereksekusi semua.

```

1 def delPr():
2     try:
3         showPr()
4         No = int(input("Masukkan No produk yang ingin dihapus: "))
5
6         if No in produk:
7             del produk[No]
8             print("Produk berhasil dihapus!")
9         else:
10            print("Produk tidak ditemukan!")
11            input("Tekan Enter untuk melanjutkan...")
12    except ValueError:
13        print("No produk harus berupa angka!")
14        input("Tekan Enter untuk melanjutkan...")
15    except Exception as e:
16        print(f"Terjadi kesalahan: {str(e)}")
17        input("Tekan Enter untuk melanjutkan...")
18
19 def buy(username):
20     try:
21         showPr()
22         No = int(input("Masukkan No produk yang akan dibeli: "))
23         jumlah = int(input("Masukkan jumlah yang akan dibeli: "))
24
25         if No in produk:
26             if produk[No]["stok"] >= jumlah:
27                 total = produk[No]["harga"] * jumlah
28                 confirmation = input(f"Total pembayaran: Rp{total:,}. Apakah anda yakin? (y/n): ")
29                 if confirmation.lower() == 'y':
30                     id_transaksi = len(listTerjual) + 1
31                     listTerjual.append({
32                         "id": id_transaksi,
33                         "username": username,
34                         "produk_id": No,
35                         "jumlah": jumlah,
36                         "total": total
37                     })
38                     produk[No]["stok"] -= jumlah
39                     print(f"Pembelian berhasil!\nTotal: Rp{total:,}\nDitunggu Repeat ordernya!")
40                     input("Tekan Enter untuk melanjutkan...")
41             else:
42                 print("Stok tidak mencukupi!")
43                 input("Tekan Enter untuk melanjutkan...")
44         else:
45             print("Produk tidak ditemukan!")
46             input("Tekan Enter untuk melanjutkan...")
47    except ValueError:
48        print("No produk dan jumlah harus berupa angka!")
49        input("Tekan Enter untuk melanjutkan...")
50    except Exception as e:
51        print(f"Terjadi kesalahan: {str(e)}")
52        input("Tekan Enter untuk melanjutkan...")

```

Lalu pada bagian yang ini saya membuat kode bagian menghapus dan dibawahnya saya melakukan *def* pada bagian membeli. Pada bagian “delPr()” saya meminta *user* untuk melakukan *input* nomor produk mana yang ingin di hapus.

Lalu saya menggunakan “if No in produk” untuk melakukan pengecekan apakah nomor yang di *input* oleh *user* tersedia di *dictionary* atau tidak. Lalu saya menggunakan “del produk[No]” yang berarti akan melakukan penghapusan pada produk yang kita *input* tadi.

```

1 def terjual():
2     clear()
3     print("=" * 9, "DAFTAR PENJUALAN", "=" * 9)
4     try:
5         if listTerjual:
6             df = pd.DataFrame(listTerjual)
7             df['Produk'] = df['produk_id'].map(lambda x: produk.get(x, {}).get('nama', 'Produk Dihapus'))
8             print(df[['id', 'username', 'Produk', 'jumlah', 'total']].to_string(index=False))
9         else:
10            print("Belum ada Transaksi")
11    except Exception as e:
12        print(f"Terjadi kesalahan: {str(e)}")
13        input("\nTekan Enter untuk melanjutkan...")
14
15 def riwayatUser(username):
16     clear()
17     print(f"=" * 7, f"RIWAYAT PEMBELIAN {username}", "=" * 7)
18
19     try:
20         userlistTerjual = [sale for sale in listTerjual if sale['username'] == username]
21
22         if userlistTerjual:
23             print(f"{'ID':<5} {'Produk':<20} {'Jumlah':<10} {'Total (IDR)':<15}")
24             print("=" * 50)
25
26             for sale in userlistTerjual:
27                 namaProduk = produk.get(sale['produk_id'], {}).get('nama', 'Produk Dihapus')
28                 print(f"{sale['id']:<5} {namaProduk:<20} {sale['jumlah']:<10} {sale['total']:<15}")
29             else:
30                 print("Belum ada pembelian.")
31    except Exception as e:
32        print(f"Terjadi kesalahan: {str(e)}")
33        input("\nTekan Enter untuk melanjutkan...")
34

```

Pada lines 6 saya mengubah listTerjual ke DataFrame, DataFrame ini memudahkan kita untuk memanipulasi data. Kemudian saya menggunakan “produk.get” yang berfungsi untuk mencari nama produk berdasarkan produk ID dengan permisalan “x”, jika *program* mendapatkan *data* produk, kita mendapatkan data produk tersebut. Namun jika tidak, kita akan mengembalikannya ke dalam “{ }” sebagai default.

Kemudian jika *program* tidak dapat menemukan produk dengan produkID yang ada, maka akan mengembalikan “Produk dihapus”.

Saya menggunakan “try” dan “except” hampir pada keseluruhan kode agar dapat mengetahui error apa yang terjadi jika ada kesalahan.

```

1 def adminMenu():
2     load()
3     while True:
4         try:
5             clear()
6             print("=" * 25, "MENU ADMIN", "=" * 25, "\n1. Tambah Produk\n2. Lihat Produk\n3. Update Produk\n4. Hapus Produk\n5. Lihat Penjualan\n0. Back")
7
8             pilih = input("Pilih Menu : ")
9
10            if pilih == "1":
11                addPr()
12            elif pilih == "2":
13                showPr()
14                input("Tekan Enter untuk melanjutkan...")
15            elif pilih == "3":
16                updPr()
17            elif pilih == "4":
18                delPr()
19            elif pilih == "5":
20                terjual()
21            elif pilih == "0":
22                break
23            else:
24                print("Pilihan tidak valid!")
25                input("Tekan Enter untuk melanjutkan...")
26        except Exception as e:
27            print(f"Terjadi kesalahan: {str(e)}")
28            input("Tekan Enter untuk melanjutkan...")
29
30 def userMenu(username):
31     load()
32     while True:
33         try:
34             clear()
35             print("=" * 7, f"Hi {username} Apa yang ingin anda lakukan?", "=" * 7, "\n1. Lihat Produk\n2. Beli Produk\n3. History Pembelian\n4. Total Belanja\n0. Back")
36
37             pilih = input("Pilih menu : ")
38
39             if pilih == "1":
40                 showPr()
41                 input("Tekan Enter untuk melanjutkan...")
42             elif pilih == "2":
43                 buy(username)
44             elif pilih == "3":
45                 riwayatUser(username)
46             elif pilih == "4":
47                 lihatTotalBelanja(username)
48             elif pilih == "0":
49                 break
50             else:
51                 print("Pilihan tidak valid!")
52                 input("Tekan Enter untuk melanjutkan...")
53         except Exception as e:
54             print(f"Terjadi kesalahan: {str(e)}")
55             input("Tekan Enter untuk melanjutkan...")

```

Disini saya membuat bagian yang akan ditampilkan pada “Menu User” dan “Menu Admin” saya membuatnya berbeda agar memberi batasan pada user sehingga tidak terjadi hal-hal yang tidak diinginkan.

```
1 def totalBelanja(index, username):
2     if index >= len(listTerjual):
3         return 0
4     transaksi = listTerjual[index]
5     totalR = totalBelanja(index + 1, username)
6
7     if transaksi['username'] == username: #menambahkan jika username sesuai
8         return transaksi['total'] + totalR
9
10    return totalR
11
12 def lihatTotalBelanja(username):
13     clear()
14     print(f"=" * 7, f"TOTAL BELANJA {username}", "=" * 7)
15     total = totalBelanja(0, username)
16     print(f"\nTotal belanja Anda: Rp{total:,}")
17     input("\nTekan Enter untuk melanjutkan...")
```

Disini saya melakukan fungsi dengan parameter untuk menghitung total belanja, jadi *program* akan melakukan pengecekan semua index sesuai dengan *username* yang *login* di *program* pada saat itu. Dan setelah itu *program* akan melakukan penjumlahan semua total belanja sesuai dengan *username* nya.

```

1 import sys
2 import time
3 from login import login, register, clear
4 from sistem import adminMenu, userMenu
5
6 def main():
7     while True:
8         clear()
9         print("-" * 11, "Main Menu", "-" * 11, "\n1. Login\n2. Register\n0. Keluar")
10
11         pilih = input("\nPilih Menu: ")
12
13         if pilih == "1":
14             user = login()
15             if user:
16                 if user["role"] == "admin":
17                     adminMenu()
18                 else:
19                     userMenu(user["username"])
20         elif pilih == "2":
21             register()
22         elif pilih == "0":
23             clear()
24             kalimat = "Terimakasih sudah menggunakan Sistem kami:\n\tHave a nice day!"
25             for char in kalimat:
26                 sys.stdout.write(char)
27                 time.sleep(0.03)
28             break
29         else:
30             print("Pilihan tidak valid!")
31             input("Tekan Enter untuk melanjutkan...")
32
33 if __name__ == "__main__":
34     main()
35

```

Terakhir kita berada pada *file* “Main” saya melakukan *import* hal-hal yang saya perlukan agar tidak perlu menulis ulang. Saya melakukan “def main()” Sebagai menu utamanya, dan untuk membedakan *role* diantara *user* saya menggunakan “if user[“role”] == “admin” ” yang berarti pada *dictionary users* bagian “role” harus bertuliskan “admin” agar “adminMenu()” dijalankan.

Diakhir saya menggunakan “if __name__ == “__main__” agar menandakan bahwa ini adalah *file* utamanya, dan menghindari hal-hal yang tidak diinginkan.

```

----- Main Menu -----
1. Login
2. Register
0. Keluar

Pilih Menu: █

```

Jika kita menjalankan *program* hal pertama yang keluar adalah “Main Menu”

```

===== LOGIN =====
Username: dsa
Password: das
Username atau password salah! Kembali ke menu utama.
Tekan Enter untuk melanjutkan...█

```

Ini hal yang terjadi jika kita salah dalam melakukan *input username* dan *password*. Dan *program* akan kembali ke “Main Menu”

```

===== REGISTER =====
Username baru: tes
Username sudah digunakan!
Tekan Enter untuk melanjutkan...█

```

Berikut hal yang terjadi jika *user* melakukan *input username* yang sama pada saat *register*.

```

===== REGISTER =====
Username baru: ucup
Password: ucuppower
Succes!
Tekan Enter untuk melanjutkan...█

```

Dan ini yang akan terjadi jika *user* berhasil melakukan *register*

```

===== LOGIN =====
Username: admin
Password: admin123
Loading: [=====] 36%█

```

Setelah berhasil *login*, *user* maupun *admin* akan mendapatkan *loading* seperti ini sebelum lanjut ke *program* utamanya.

```

===== MENU ADMIN =====
1. Tambah Produk
2. Lihat Produk
3. Update Produk
4. Hapus Produk
5. Lihat Penjualan
0. Back
Pilih Menu : █

```

Ini adalah tampilan untuk *user* yang memiliki *role* “Admin”

```

===== TAMBAH PRODUK =====
Nama produk: Cushion
Harga produk: 100000
Stok produk: 20
Produk berhasil ditambahkan!
Tekan Enter untuk melanjutkan...

```

Ketika *admin* ingin menambahkan produk, *admin* akan diminta melakukan *input* Nama produk, Harga produk, dan Stok produk untuk ditambahkan ke tabel daftar produk.

```

----- DAFTAR PRODUK -----

```

No	Nama	Harga (IDR)	Stok
1	Facewash	200,000	50
2	Liptint	500,000	50
3	Serum	500,000	30
4	Cushion	100,000	20

```

Tekan Enter untuk melanjutkan...

```

Dan Produk yang kita tambahkan tadi otomatis akan tertambah ke dalam “Daftar Produk” sesuai dengan yang kita isi tadi.

```

===== Hi ucap Apa yang ingin anda lakukan? =====
1. Lihat Produk
2. Beli Produk
3. History Pembelian
4. Total Belanja
0. Back
Pilih menu :

```

Ini adalah menu dari *user* jika *role* nya bukan *admin*. Tampilan pada kata setelah “hi” akan mengikut dengan *username* dari *user*.


```
----- DAFTAR PRODUK -----  
  
No      Nama      Harga (IDR)      Stok  
=====
```

1	Facewash	200,000	50
2	Liptint	500,000	50
3	Serum	500,000	30
4	Cushion	100,000	20

```
Masukkan No produk yang akan dibeli: 1  
Masukkan jumlah yang akan dibeli: 20  
Total pembayaran: Rp4,000,000. Apakah anda yakin? (y/n): y  
Pembelian berhasil!  
Total: Rp4,000,000  
Ditunggu Repeat ordernya!  
Tekan Enter untuk melanjutkan...
```

Jika *user* ingin melakukan pembelian, *user* diminta untuk melakukan *input* no produk dan jumlah yang ingin dibeli, serta *user* akan diberi konfirmasi ulang apakah ingin melanjutkan atau tidak pembeliannya. Jika kita melakukan *input* selain “y” maka program akan balik ke menu kedua tadi, yaitu bagian “menu user”.

```
===== RIWAYAT PEMBELIAN ucup =====  
ID      Produk      Jumlah      Total (IDR)  
=====
```

1	Facewash	20	4,000,000
---	----------	----	-----------

```
Tekan Enter untuk melanjutkan...
```

Disini *user* dapat melihat riwayat pembelian yang sudah dilakukan

```
===== TOTAL BELANJA ucup =====  
  
Total belanja Anda: Rp4,000,000  
  
Tekan Enter untuk melanjutkan...
```

Berikut adalah *output* dari total belanja ucup

```
===== DAFTAR PENJUALAN =====  
id username  Produk  jumlah  total  
1    ucup Facewash    20 4,000,000  
  
Tekan Enter untuk melanjutkan...
```

Dan ketika *user* tadi melakukan pembelian akan otomatis masuk juga pada “Daftar Penjualan” pada *admin*.

```
Terimakasih sudah me
```

```
Terimakasih sudah menggunakan Sistem kami:)  
Have a nice day!
```

Dan ketika *user* ingin keluar dari program, akan terjadi hal seperti ini. Ini adalah fungsi dari *sys* tadi yang berfungsi menulis kalimat menjadi huruf per huruf hingga seluruh susunan kata ter *print*.

```
=== MENU ADMIN ===  
1. Tambah Produk  
2. Lihat Produk  
3. Update Produk  
4. Hapus Produk  
5. Lihat Penjualan  
6. Keluar  
Pilih menu (1-6): 3  
Terjadi kesalahan: name 'showProdukroduk' is not defined  
Tekan Enter untuk melanjutkan . . .
```

Dan ini fungsi dari “except exception as e:” dapat mendeteksi kesalahan dari kode yang saya buat, sehingga tidak langsung berhenti saat terjadi *error*.