

Simple Transaction Program With Python

Probability Course - Sekolah Data Pacmann

Outline

- Background
- Requirements
- Workflow
- Solution
- Test Case
- Conclusion

Background

Background

- Andi adalah seorang pemilik supermarket besar di salah satu kota di Indonesia. Andi memiliki rencana untuk melakukan ekspansi bisnis, yaitu Andi akan membuat sistem kasir yang self-service di supermarket miliknya. Sehingga, customer bisa langsung memasukkan item yang dibeli, jumlah item yang dibeli, dan harga item yang dibeli dan fitur yang lain. Sehingga, customer yang tidak berada di kota tersebut bisa membeli barang dari supermarket tersebut.
- Setelah Andi melakukan riset, ternyata Andi memiliki masalah, yaitu Andi membutuhkan Programmer untuk membuatkan fitur-fitur agar bisa sistem kasir self-service di supermarket itu bisa berjalan dengan lancar.

Requirements

Requirements

Membuat fitur dimana user bisa melakukan hal-hal berikut:

- Membuat ID transaksi
- Memasukan daftar barang ke daftar belanjaan
- Update atau edit daftar belanjaan
- Menghapus barang pada daftar belanjaan
- Reset transaksi
- Melihat/mengecek daftar belanjaan
- Menghitung total harga daftar belanjaan

Requirements

Untuk bisa menjalankan fitur-fitur tersebut, maka diperlukan data berikut ini:

- Nama barang
- Harga barang per satuan
- Satuan

Workflow

Workflow



Solution

Proposed Solutions

Membuat fitur dimana user bisa melakukan hal-hal berikut:

- Membuat ID transaksi
- Memasukan daftar barang ke daftar belanjaan
- Update atau edit daftar belanjaan
- Menghapus barang pada daftar belanjaan
- Reset transaksi
- Melihat/mengecek daftar belanjaan
- Menghitung total harga daftar belanjaan
- (Tambah) Menampilkan daftar barang yang tersedia di supermarket

Proposed Solutions

Membuat ID transaksi

- Membuat object class untuk menginisiasi transaksi
- User menginisiasi ID transaksi
 - Menggunakan method
`transaction_n = Transaction()`

Proposed Solutions

Memasukan daftar barang ke daftar belanjaan

- User menginisiasi method
transaction_n.add_to_cart()
- Parameter yang diinput
 - Barang yang mau dibeli
 - Jumlah yang mau dibeli

Proposed Solutions

Update atau edit daftar belanjaan

- User menginisiasi method
transaction_n.update_cart()
- Parameter yang diinput
 - Nomer barang yang mau diganti
 - Barang baru yang mau dibeli
 - Jumlah barang baru yang mau dibeli

Proposed Solutions

Menghapus barang pada daftar belanjaan

- User menginisiasi method
transaction_n.delete_item()
- Ketika diinisiasi, menjalankan fungsi pop()
- Parameter yang diinput
 - Nomer barang yang mau dihapus

Proposed Solutions

Reset transaksi

- User menginisiasi method
transaction_n.reset_order()
- Ketika diinisiasi, menjalankan fungsi input string untuk memastikan reset order
- Parameter yang diinput
 - String (Ya/Tidak)

Proposed Solutions

Melihat/mengecek daftar belanjaan

- User menginisiasi method
transaction_n.check_order()
- Ketika diinisiasi, menampilkan daftar belanjaan, sekaligus sudah termasuk perhitungan total order beserta diskonnya
- Parameter yang ditampilkan:
 - Nomer barang
 - Nama barang
 - Jumlah
 - Harga per satuan
 - Harga Total
 - Harga total belanjaan
 - Diskon
 - Harga total belanjaan setelah diskon

Proposed Solutions

Menghitung total harga daftar belanjaan

- Fitur ini turut disertakan bersama fitur melihat/mengecek daftar belanjaan

Proposed Solutions

(Tambahan) Menampilkan daftar barang yang tersedia di supermarket

- User menginisiasi method:
`transaction_n.check_item()`
- Fitur ini berguna dalam memudahkan user memasukan barang yang dicarinya

Implemented code

Script dapat dilihat pada repository berikut:

<https://github.com/Widyamuktiwaskito/final-project-python>

Test Case

Test case 1

Customer ingin menambahkan dua item baru.

```
[ ] transaction_1.add_to_cart()
```

Masukan data belanjaan Anda
Barang: Beras
Jumlah: 2
Harga Satuan: Rp 11000 per 1kg
Total Harga: Rp 22000

```
[ ] transaction_1.add_to_cart()
```

Masukan data belanjaan Anda
Barang: Ayam
Jumlah: 3
Harga Satuan: Rp 32000 per 1 ekor
Total Harga: Rp 96000

```
[ ] transaction_1.check_order()
```

Data Transaksi:

No	Barang	Jumlah	Harga Satuan	Harga Total
1	Beras	2	Rp 11000	Rp 22000
2	Ayam	3	Rp 32000	Rp 96000

Total pesanan adalah: Rp 118000

Besaran diskon adalah: Rp 0

Total pesanan setelah diskon: Rp 118000

Pemesanan sudah benar

Test case 2

Customer ingin menghapus salah satu barang dari daftar belanjaannya.

```
[ ] transaction_1.delete_item()
```

```
Hapus data belanjaan Anda
Data Transaksi:
| No | Barang | Jumlah | Harga Satuan | Harga Total |
|----|-----|-----|-----|-----|
| 1 | Beras | 2 | Rp 11000 | Rp 22000 |
| 2 | Ayam | 3 | Rp 32000 | Rp 96000 |
Total pesanan adalah: Rp 118000
Besaran diskon adalah: Rp 0
Total pesanan setelah diskon: Rp 118000
Pemesanan sudah benar
Masukan nomer barang di daftar belanjaan Anda yang ingin dihapus.
Nomer Barang: 2
```

```
[ ] transaction_1.check_order()
```

```
Data Transaksi:
| No | Barang | Jumlah | Harga Satuan | Harga Total |
|----|-----|-----|-----|-----|
| 1 | Beras | 2 | Rp 11000 | Rp 22000 |
Total pesanan adalah: Rp 22000
Besaran diskon adalah: Rp 0
Total pesanan setelah diskon: Rp 22000
Pemesanan sudah benar
```

Test case 3

Customer ingin mereset dan mengulang belanjaannya.

```
[ ] transaction_1.reset_order()
```

```
Data Transaksi:
| No | Barang | Jumlah | Harga Satuan | Harga Total |
|----|-----|-----|-----|-----|
| 1  | Beras  | 2      | Rp 11000   | Rp 22000    |
Total pesanan adalah: Rp 22000
Besaran diskon adalah: Rp 0
Total pesanan setelah diskon: Rp 22000
Pemesanan sudah benar
Anda ingin reset order (Ya/Tidak)?Ya
```

```
[ ] transaction_1.check_order()
```

```
Data Transaksi:
| No | Barang | Jumlah | Harga Satuan | Harga Total |
|----|-----|-----|-----|-----|
Total pesanan adalah: Rp 0
-----
UnboundLocalError                                                 Traceback (most recent call last)
<ipython-input-156-f2b878de653d> in check_order(self)
      99         print(f"Total pesanan adalah: Rp {total_order}")
--> 100         print(f"Besaran diskon adalah: Rp {diskon}")
     101         print(f"Total pesanan setelah diskon: Rp {harga_total_setelah_diskon}")

UnboundLocalError: local variable 'diskon' referenced before assignment
```

During handling of the above exception, another exception occurred:

```
Exception                                                 Traceback (most recent call last)
----- 1 frames -----
<ipython-input-156-f2b878de653d> in check_order(self)
    102         print(f"Pemesanan sudah benar")
    103     except:
--> 104         raise Exception(f"Order masih 0.")
    105
    106     # HAPUS BARANG DI ORDER
```

```
Exception: Order masih 0.
```

Test case 4

Customer ingin mengetahui total harga belanjaannya.

```
[ ] transaction_1.check_order()
```

Data Transaksi:

No	Barang	Jumlah	Harga Satuan	Harga Total
1	Beras	5	Rp 11000	Rp 55000
2	Ayam	9	Rp 32000	Rp 288000
3	Kangkung	15	Rp 8000	Rp 120000

Total pesanan adalah: Rp 463000

Besaran diskon adalah: Rp 37040.0

Total pesanan setelah diskon: Rp 425960.0

Pemesanan sudah benar

Conclusion

Conclusion

- Perancangan program kasir self-service ini tentunya masih banyak kekurangan. Namun begitu, hal tersebut membuka peluang untuk pengembangan selanjutnya yang dapat berguna bagi pengalaman user dalam bertransaksi.
- Dari identifikasi saya sementara, fitur yang dapat dikembangkan selanjutnya adalah:
 - Main menu
 - Admin
 - Storing ID transaksi
- Tentunya, dengan peningkatan pengetahuan akan python, terutama object oriented programming, akan dapat mengembangkan program ini menjadi lebih tepat guna.

Thank you!
