

Python Project

Python, Sekolah Data, Pacmann

Outline

Background Project	2
Feature Requirements	3
Petunjuk Pengerjaan	5
Langkah #1 - Pahami Latar Belakang Problem s	5
Langkah #2 - Membuat Requirements atau Objectives	5
Langkah #3 - Membuat Code berdasarkan Requirements yang ditentukan	5
Langkah #4 - Mengubah Code menjadi Script	6
Langkah #5 - Coba Code dengan "Test Case"	6
Langkah #6 - Upload Project ke Github dan berikan Dokumentasi	7
Outcome Project	8
Dokumentasi (Github)	8
Presentasi	8
Evaluasi	9
Need Assistance?	10
Tools	11
Format Dokumentasi	12
Requirements / Objectives	12
Alur Program / Flowchart	12
Penjelasan Code	12
Hasil Test Case	13
Conclusion/Future Work	13

Dibuat dengan 

Project Python - Sekolah Data Pacmann



Background Project

Andi adalah seorang pemilik supermarket besar di salah satu kota di Indonesia. Andi memiliki rencana untuk melakukan perbaikan proses 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 mem buat fitur - fitur agar bisa sistem kasir self-service di supermarket itu bisa berjalan dengan lancar.

Feature Requirements

Akhirnya Andi meminta tolong kepada teman - teman selaku programmer Python untuk membuat program yang menyelesaikan problem tersebut.

Jika ada yang berbelanja, begini journey Customer dalam membantu orang yang berbelanja tersebut

- Customer membuat ID transaksi customer berikut
 - Dengan membuat objek dari class: `trnsct_123 = Transaction()`
- Kemudian, Custom er memasukkan nama item , jumlah item , dan harga barang.
 - Masukkan item yang ingin dibeli.
`add_item([<nama item>, <jumlah item>, <harga per item>])`
- Jika ternyata ada kesalahan dalam memasukkan nama item atau jumlah item atau harga item tetapi tidak ingin menghapus itemnya, Customer bisa melakukan
 - Update nama item dengan method:
`update_item_name(<nama item>, <update nama item>)`
 - Update jumlah item dengan method:
`update_item_qty(<nama item>, <update jumlah item>)`
 - Update harga item menggunakan method:
`update_item_price(<nama item>, <update harga item>)`
- Jika batal mem beli item belanjaan, Custom er bisa melakukan
 - Menghapus salah satu item dari nama item dengan method
`delete_item(<nama item>)`

Item	Jumlah Item	Harga/Item	Harga total
mobil	2	100.000	200.000
(data yang dihapus)			
tempe	3	3.000	9.000

Ketika menghapus salah satu nama item , maka jumlah item dan harga per item pada baris/list tersebut akan ikut terhapus

- Langsung menghapus sem ua transaksi atau reset transaksi dengan method `reset_transaction()`
- Customer sudah selesai dengan berbelanja online nya, tetapi Custom er masih ragu apakah harga barang dan nam a barang yang diinput sudah benar. Bisa saja Customer melakukan kesalahan dalam melakukan input, sem isal sudah melakukan input harga barang tetapi lupa untuk input nam a barangnya. Andi bisa menggunakan method `check_order()`. Terdapat ketentuan:
 - Akan mengeluarkan pesan “**Pemesanan sudah benar**” (*bebas bisa dengan message yang lain*) jika tidak ada kesalahan input

- Akan mengeluarkan pesan “**Terdapat kesalahan input data**” jika terjadi kesalahan input

- Keluarkan output transaksi atau pemesanan apa saja yang sudah dibeli.
- Contoh Output:

No	Nama Item	Jumlah Item	Harga/Item	Total Harga
1	Mobil	2	100000	200000
2	Mie	1	5000	5000
3	Tempe	3	3000	9000

- Setelah melakukan pengecekan, Custom er bisa menghitung total belanja yang sudah dibeli. Andi bisa menggunakan method `total_price()`. Pada superm arket ini ternyata terdapat ketentuan:
 - Jika total belanja Andi diatas Rp 200.000 maka akan mendapatkan diskon 5%
 - Jika total belanja Andi diatas Rp 300.000 maka akan mendapatkan diskon 8%
 - Jika total belanja Andi diatas Rp 500.000 maka akan mendapatkan diskon 10%

Andi juga memberikan pesan kepada teman - teman kalau diberi kebebasan untuk menambahkan fitur yang lain apabila masih terdapat fitur yang belum tercover dalam sistem tersebut.

Catatan: Project dikerjakan secara Individu.



Petunjuk Pengerjaan

Untuk mempermudah pengerjaan, berikut adalah hal-hal yang dapat Anda lakukan.

Langkah #1 - Pahami Latar Belakang Problems

Anda diberikan case Super Cashier, sebelum anda membuat program nya pastikan anda paham terlebih dahulu terhadap problem yang dihadapi dan apa yang harus dilakukan untuk menyelesaikan problem tersebut.

Langkah #2 - Membuat Requirements atau Objectives

Expected Time to Start: [Week 2](#)

Estimated Time of Completion: [End of Week 4](#)

Selanjutnya setelah memahami problem yang dihadapi, buatlah requirements / objectives yang harus dipenuhi untuk mempermudah membuat program. Ada nantinya. Sebagai contoh:

- Alur Program atau Flowchart
- Proses atau fungsi apa saja yang dibutuhkan untuk menyelesaikan problem tersebut
- dll

Langkah #3 - Membuat Code berdasarkan Requirements yang ditentukan

Expected Time to Start: [Week 3](#)

Estimated Time of Completion: [End of Week 6](#)

Setelah menentukan requirements atau objectives, buatlah code yang memiliki ketentuan sebagai berikut:

- Membuat fitur - fitur yang sudah disebutkan di requirements
- Modular Code
- Clean Code (PEP8)
- Terdapat dokumentasi Docstring di Function atau Class yang dibuat
- Terdapat try, error di branching agar mudah untuk melakukan track error (defense programming)
- Boleh menggunakan library atau method Python untuk mempermudah melakukan perhitungan. (NumPy, dsb)

Langkah #4 - Mengubah Code menjadi Script

Expected Time to Start: [Week 4](#)

Estimated Time of Completion: [End of Week 6](#)

Setelah membuat code berdasarkan requirements yang dibuat, ubahlah code tersebut menjadi bentuk **script (.py)**. Sehingga code tersebut bisa menjadi module atau library.

Langkah #5 - Coba Code dengan "Test Case"

Expected Time to Start: [Week 6](#)

Estimated Time of Completion: [End of Week 7](#)

Setelah mengubah code menjadi modular code, langkah selanjutnya adalah mencoba modular code tersebut di Notebook. Setelah itu anda diwajibkan untuk mencoba test case yang sudah disediakan oleh tim Pacmann. Jika output sesuai dengan yang diinginkan, maka lolos test case (jika format outputnya tidak sama tidak apa - apa, yang penting outputnya sesuai).

Test 1:

Customer ingin menambahkan dua item baru menggunakan method `add_item()`. Item yang ditambahkan adalah sebagai berikut:

- Nama Item : Ayam Goreng, Qty: 2, Harga: 20000
- Nama Item : Pasta Gigi, Qty: 3, Harga: 15000

Expected Output:

```
/mnt/d/Con/Python/Project python cashier.py
Item yang dibeli adalah: {'Ayam Goreng': [2, 20000], 'Pasta Gigi': [3, 15000]}
```

Test 2:

Dibuat dengan ❤️

Project Python - Sekolah Data Pacmann

Ternyata Customer salah membeli salah satu item dari belanjaan yang sudah ditambahkan, maka Customer menggunakan method `delete_item()` untuk menghapus item. Item yang ingin dihapuskan adalah **Pasta Gigi**.

Expected Output:

```
/mnt/d/Con/Python/Project python cashier.py
{'Ayam Goreng': [2, 20000]}
```

Test 3:

Ternyata setelah dipikir - pikir Customer salah memasukkan item yang ingin dibelanjakan! Daripada menghapusnya satu - satu, maka Customer cukup menggunakan method `reset_transaction()` untuk menghapus semua item yang sudah ditambahkan.

Expected Output:

```
/mnt/d/Con/Python/Project python cashier.py
Semua item berhasil di delete!
```

Test 4:

Setelah Customer selesai berbelanja, akan menghitung total belanja yang harus dibayarkan menggunakan method `total_price()`. Sebelum mengeluarkan output total belanja akan menampilkan item - item yang dibeli.

Expected Output:

```
/mnt/d/Con/Python/Project python cashier.py
Item yang dibeli adalah {'Ayam Goreng': [2, 20000], 'Pasta Gigi': [3, 15000], 'Mainan Mobil': [1, 200000], 'Ml Instan': [5, 3000]}
Total Belanja yang harus dibayarkan adalah Rp. 285000.0
```

Langkah #6 - Upload Project ke Github dan berikan Dokumentasi

*Expected Time to Start: **Week 7***

*Estimated Time of Completion: **End of Week 8***

Setelah membuat code dan mencobanya di test case, upload project anda ke Github dan berikan dokumentasi, yang berisi:

- Latar belakang Problem s
- Penjelasan Requirements / Objectives
 - Alur Program / Flowchart
 - Proses atau Fungsi apa yang dibutuhkan
- Snippet Code atau penjelasan masing - masing Attributes atau Functions cara kerja / fungsi nya bagaimana
- Kasih contoh kalau dijalankan output nya bagaimana (boleh dengan test case)

Dibuat dengan ❤️

Project Python - Sekolah Data Pacmann

Outcome Project

Setelah Anda mengerjakan itu semua, kami ingin Anda dapat merangkum hasilnya dalam sebuah Dokumentasi (Github) & presentasi.

Dokumentasi (Github)

- Merangkum hasil pekerjaan Anda yang sudah dibuat dalam bentuk dokumentasi.
- Berisi:
 - Latar belakang problems
 - Penjelasan requirements / objectives yang dibutuhkan untuk membuat program
 - Penjelasan alur code atau flowchart.
 - Penjelasan mengenai functions atau attribute yang dibuat dan fungsinya untuk apa
 - Demonstrasi code dengan Test Case yang diberikan dan outputnya bagaimana
 - Conclusion
- Push code dan dokumentasi ke Repo Github
- Dokumentasi dalam bentuk **README.md** bukan .pdf
- Referensi Github:
 - <https://github.com/putusetya/LMS-Project>
 - <https://github.com/ovtakurniawan/python-LMS>
- Referensi menulis Dokumentasi yang Baik dan Benar:
 - [Design Docs at Google](#)
 - [GitHub - iamleibuilds/documentation-handbook: How to write high-quality friendly](#)

- Permission
Set sebagai publik agar dapat tim Pacmann periksa.

Evaluasi

Kami akan mengevaluasi beberapa komponen berikut.

Fitur penilaian	Bobot [%]
1. Membuat requirements berdasarkan problems	10
2. Terdapat penjelasan alur program atau flowchart	10
3. Kelengkapan code yang berdasarkan requirements	25
4. Membuat modular code dengan format script	15
5. Lolos Test Case	20
6. Upload Project ke Github	
a. Hanya terdapat code nya saja	8
b. Menuliskan Dokumentasi	7
7. Membuat Video presentasi	5

Need Assistance?

Tentu project ini menantang!
Jika anda memiliki pertanyaan atau kesulitan dalam mengerjakan project ini, anda bisa memanfaatkan fasilitas Asistensi Via discord tag asisten.

Tools

Anda dibebaskan untuk menggunakan tools apa saja untuk membuat code dan alur program/flowchart.

- Python
- Jupyter Notebook / Jupyter Lab / Google Colab
- Google Docs / Diagrams

Format Dokumentasi

Requirements / Objectives

- Jelaskan requirements / objectives yang dibutuhkan / diinginkan / dicapai untuk membuat program tersebut
- Misal:
 - *Mem buat proses untuk menyelesaikan problem untuk membuat pemesanan hotel*
 - *Mem buat proses untuk menghitung total_belanja berdasarkan problem*
 - *Dll*

Alur Program / Flowchart

- Jelaskan alur program yang anda buat berdasarkan requirements.
- Bisa dalam bentuk text atau gambar (Flowchart)
- Misal:
 - Langkah pertama kita membutuhkan input xxx
 - Langkah kedua input akan masuk ke proses total_belanja untuk mendapatkan hasil akhirnya
 - dll
- Referensi untuk pembuatan Flowchart
 - [Flowchart Adalah: Fungsi, Jenis, Simbol, dan Contohnya - Dicode Blog](#)
 - [What is a Flowchart | Lucidchart](#)


Penjelasan Code

- Jelaskan script code yang dibuat itu untuk apa dan fungsinya apa
- Misal:
 - Script 'main.py' berfungsi untuk xxx dan di dalamnya terdapat function xxx untuk melakukan yyy
 - Dll
- Lalu, berikan potongan code / snippet code dari masing - masing function dan jelaskan itu untuk melakukan proses apa:
- Misal:

```
def total_belanja(list_belanja):
    ...

    Function untuk menghitung total belanja yang dimiliki berdasarkan
    barang belanjaan yang dibeli.
```

Parameters

Dibuat dengan 

Project Python - Sekolah Data Pacmann

```
list_belanja: list
    Berisi harga - harga barang yang dibeli
xxx
...

return sum(list_belanja)
```

Hasil Test Case

- Jelaskan test case ini itu untuk mencoba proses atau function yang sudah dibuat
- Berikan output nya berdasarkan test case yang sudah diberikan
- Hasil test case bisa dalam bentuk potongan code dan output nya seperti apa atau langsung dalam bentuk screenshot gambar. (bisa cek di Github referensi di atas)

Conclusion/Future Work

- Rangkum kembali apa saja yang sudah dikerjakan
- Untuk future work, m ari kita berandai - andai, jika anda memiliki waktu lebih dan SDM lebih, apa yang akan Anda improve dalam program yang sudah dibuat?

Dibuat dengan 

Project Python - Sekolah Data Pacmann