

Dasar-Dasar Pemrograman 1 Gasal 2023/2024

Lab 08

OOP Dasar



FAKULTAS
ILMU
KOMPUTER

Paciloka

Sub-CPMK 9 : Mampu menganalisis dan merancang classes dalam konteks *object-oriented programming* secara sederhana



source:

<https://dikti.kemdikbud.go.id/kabar-dikti/kampus-kita/fasilkom-ui-luncurkan-aplikasi-web-program-magang-kampus-merdeka/>

Setelah Pak Esde berhasil membuat program yang mencatat silsilah keluarga seseorang, Pak Esde ingin membuka bisnis OTA (Online Travel Agent) untuk membantu pemesanan hotel secara cepat dan mudah. Pak Esde kembali meminta Dek Depe untuk membuat programnya, tetapi melihat Dek Depe yang mulai kewalahan dengan permintaan Pak Esde, kamu sebagai teman yang baik pun berniat menolongnya.

Ketentuan Program

- **Keterangan**

Kalian akan diberikan template [paciloka template.py](#) untuk mempermudah penggerjaan, tetapi **tidak wajib digunakan**.

Program **harus memiliki** class beserta atribut sebagai berikut:

- **Class Hotel**
 - **name** : nama hotel
 - **available_room** : banyaknya kamar hotel yang tersedia
 - **room_price** : harga satu kamar hotel. Asumsikan semua harga kamar di hotel tersebut sama
 - **profit** : hasil penjualan *booking* kamar hotel
 - **guest** : daftar nama *user* yang telah melakukan *booking* kamar di hotel tersebut
- **Class User**
 - **name** : nama *user*
 - **money** : saldo *user*
 - **hotel_list** : daftar nama hotel yang pernah di-*booking* oleh user

Untuk **Class Hotel**, terdapat beberapa function yang harus diimplementasikan, yaitu :

- **__init__** : Inisialisasi Class Hotel
- **booking** : Fungsi yang akan dipanggil ketika *booking* kamar hotel
- **__str__** : Fungsi yang mengkonversi *object* Hotel menjadi *string*

Untuk **Class User**, terdapat beberapa function yang harus diimplementasikan, yaitu :

- **__init__** : Inisialisasi Class User
- **topup** : Fungsi yang akan dipanggil ketika user melakukan *topup*
- **__str__** : Fungsi yang mengkonversi *object* User menjadi *string*

Mahasiswa dipersilahkan untuk menambahkan *function* lain apabila diperlukan.

Alur Program

Di awal, program akan meminta input berupa :

1. **Banyak hotel (int)** : jumlah hotel yang ada di sistem, dipastikan valid (lebih dari 0)
2. **Banyak user (int)** : jumlah user yang ada di sistem, dipastikan valid (lebih dari 0)

Kemudian untuk setiap hotel, minta input berupa :

1. **Nama hotel (string)**, dipastikan unik (tidak ada di dalam sistem sebelumnya) dan bersifat case sensitive
2. **Jumlah kamar yang ada (int)**, dipastikan valid (lebih dari 0)
3. **Harga setiap kamar (int)**, dipastikan valid (lebih dari 0)

Kemudian untuk setiap user, minta input berupa :

1. **Nama user (string)**, dipastikan unik (tidak ada di dalam sistem sebelumnya) dan bersifat case sensitive
2. **Saldo awal user (int)**, dipastikan valid (lebih dari 0)

Lalu, program akan terus meminta input perintah sebagai berikut :

- **Daftar Perintah**

1. **Cetak semua nama hotel dan user yang ada pada sistem**

Perintah ini **diwakili dengan angka 1**. Mahasiswa diminta mencetak seluruh nama hotel dan seluruh nama user yang ada di dalam sistem dengan format yang sudah ditentukan. **Hasil pencetakan tidak harus terurut.**

2. **Cetak profit dari sebuah hotel <nama_hotel>**

Perintah ini **diwakili dengan angka 2**. Mahasiswa diminta untuk mencetak profit dari <nama_hotel>. Input <nama_hotel> tidak dijamin ada di dalam sistem.

Jika <nama_hotel> ditemukan, cetak profit hotel sesuai dengan format yang sudah ditentukan. Jika <nama_hotel> tidak ditemukan, cetak pesan error sesuai dengan format yang sudah ditentukan.

3. Seorang user <nama_user> ingin mengetahui saldoanya

Perintah ini diwakili dengan angka **3**. Mahasiswa diminta untuk mencetak saldo dari <nama_user>. Input <nama_user> tidak dijamin ada di dalam sistem.

Jika <nama_user> ditemukan, cetak saldo user sesuai dengan format yang sudah ditentukan. Jika <nama_user> tidak ditemukan, cetak pesan error sesuai dengan format yang sudah ditentukan,

4. Seorang user <nama_user> melakukan top up sebanyak <saldo>

Perintah ini diwakili dengan angka **4**. Mahasiswa diminta untuk menambahkan <saldo> untuk <nama_user>. Input <nama_user> tidak dijamin ada di dalam sistem.

Jika <nama_user> ditemukan, program akan meminta input <saldo> yang dipastikan berupa integer. Program hanya akan menambahkan saldo ke <nama_user> apabila <saldo> lebih besar dari 0 (> 0). Apabila <saldo> yang dimasukkan valid, program akan menambahkannya ke saldo <nama_user> dan mencetak output sesuai dengan format yang sudah ditentukan. Apabila <saldo> tidak valid, cetak pesan *error* sesuai dengan format yang sudah ditentukan.

Jika <nama_user> tidak ditemukan, cetak pesan *error* sesuai dengan format yang sudah ditentukan.

5. Seorang user <nama_user> melakukan booking hotel <nama_hotel> sebanyak <jumlah_kamar> kamar

Perintah ini diwakili dengan angka **5**. Mahasiswa diminta untuk melakukan input <nama_user> dan <nama_hotel>. Input <nama_user> dan <nama_hotel> tidak dijamin ada di dalam sistem.

Jika <nama_user> dan <nama_hotel> ditemukan, program akan meminta input <jumlah_kamar> (dijamin berupa integer). Kemudian, program akan mengecek apakah <nama_user> bisa melakukan booking sejumlah <jumlah_kamar> pada hotel <nama_hotel>.

Proses *booking* akan berhasil jika:

1. <nama_user> ditemukan di sistem
2. <nama_hotel> ditemukan di sistem
3. <jumlah_kamar> lebih dari 0
4. Jumlah kamar yang tersedia pada hotel tidak lebih dari <jumlah_kamar> yang dipesan <nama_user>
5. Saldo <nama_user> cukup untuk melakukan booking. Jumlah harga yang perlu dibayarkan adalah <jumlah_kamar> * harga kamar di <nama_hotel>

Apabila proses booking berhasil, maka :

1. Profit hotel akan bertambah sesuai dengan <jumlah_kamar> * harga kamar di <nama_hotel>
2. Saldo user akan berkurang sejumlah <jumlah_kamar> * harga kamar di <nama_hotel>
3. Jumlah kamar yang tersedia pada <nama_hotel> akan berkurang sejumlah <jumlah_kamar>
4. Nama hotel akan ditambahkan pada daftar hotel <nama_user> apabila tidak ada sebelumnya
5. Nama user akan ditambahkan pada daftar tamu <nama_hotel> apabila tidak ada sebelumnya

Perhatikan bahwa meskipun user memesan kamar lebih dari 1, nama user tersebut tetap **hanya ditulis satu kali** pada daftar tamu hotel tersebut.

6. Cetak semua <nama_user> yang pernah melakukan booking pada hotel <nama_hotel>

Perintah ini diwakili dengan angka **6**. Mahasiswa diminta untuk mencetak semua nama user yang telah melakukan booking pada <nama_hotel>. Input <nama_hotel> tidak dijamin ada di dalam sistem.

Jika <nama_hotel> ditemukan, cek apakah sudah ada user yang melakukan booking pada hotel tersebut. Jika tidak ada, cetak pesan error sesuai dengan format yang sudah ditentukan. Jika ada, cetak nama hotel dan seluruh nama user yang pernah pernah melakukan booking pada hotel tersebut sesuai dengan format yang sudah ditentukan. Jika <nama_hotel> tidak ditemukan, cetak pesan error sesuai dengan format yang sudah ditentukan. **Hasil pencetakan tidak harus terurut.**

7. Cetak semua nama hotel yang pernah di-booking oleh user <nama_user>

Perintah ini diwakili dengan angka **7**. Mahasiswa diminta untuk mencetak semua nama hotel yang telah di-booking oleh <nama_user>. Input <nama_user> tidak dijamin ada di dalam sistem.

Jika <nama_user> ditemukan, cek apakah user pernah melakukan booking pada suatu hotel. Jika tidak pernah, cetak pesan *error* sesuai dengan format yang sudah ditentukan. Jika pernah, cetak nama user dan seluruh nama hotel yang pernah pernah di-booking oleh user tersebut sesuai dengan format yang sudah ditentukan. Jika <nama_user> tidak ditemukan, cetak pesan *error* sesuai dengan format yang sudah ditentukan. **Hasil pencetakan tidak harus terurut.**

8. Keluar dari program Paciloka

Perintah ini diwakili dengan angka **8**. Program akan mencetak pesan selamat tinggal sesuai dengan format yang sudah ditentukan dan program akan diberhentikan 😊

Untuk ketentuan keluaran yang harus dikeluarkan dapat dilihat pada [Test Cases](#)

- **Mahasiswa WAJIB mengimplementasikan Class Hotel dan Class User untuk lab 8 ini.**

Contoh Test Cases:

Input hanya untuk memudahkan pengujian program, hasil input dan output disesuaikan dengan Output.

Test Case 1

```
Masukan banyak hotel : 1
Masukan banyak user : 1

Masukan nama hotel ke-1 : Hotel Ceria
Masukan banyak kamar hotel ke-1 : 4
Masukan harga satu kamar hotel ke-1 : 800000

Masukan nama user ke-1 : Asep
Masukan saldo user ke-1 : 1500000

=====Welcome to Paciloka=====

Masukkan perintah : 1
Daftar Hotel
1. Hotel Ceria

Daftar User
1. Asep

=====Welcome to Paciloka=====

Masukkan perintah : 2
Masukkan nama hotel : Hotel Ceria
Hotel dengan nama Hotel Ceria mempunyai profit sebesar 0

=====Welcome to Paciloka=====

Masukkan perintah : 3
Masukkan nama user : Asep
User dengan nama Asep mempunyai saldo sebesar 1500000

=====Welcome to Paciloka=====

Masukkan perintah : 4
Masukkan nama user : Asep
Masukkan jumlah uang yang akan ditambahkan ke user : 500000
Berhasil menambahkan 500000 ke user Asep. Saldo user menjadi 2000000
```

=====Welcome to Paciloka=====

Masukkan perintah : 5

Masukkan nama user : Asep

Masukkan nama hotel : Hotel Ceria

Masukkan jumlah kamar yang akan dibooking : 2

User dengan nama Asep berhasil melakukan booking di hotel Hotel Ceria dengan jumlah 2 kamar!

=====Welcome to Paciloka=====

Masukkan perintah : 6

Masukkan nama hotel : Hotel Ceria

Hotel Ceria | Asep

=====Welcome to Paciloka=====

Masukkan perintah : 7

Masukkan nama user : Asep

Asep | Hotel Ceria

=====Welcome to Paciloka=====

Masukkan perintah : 0

Perintah tidak diketahui! Masukkan perintah yang valid

=====Welcome to Paciloka=====

Masukkan perintah : 8

Terima kasih sudah mengunjungi Paciloka!

Test Case 2

```
Masukan banyak hotel : 1  
Masukan banyak user : 1
```

```
Masukan nama hotel ke-1 : Wisma Gaung  
Masukan banyak kamar hotel ke-1 : 3  
Masukan harga satu kamar hotel ke-1 : 800000
```

```
Masukan nama user ke-1 : Dek depe  
Masukan saldo user ke-1 : 2000000
```

```
=====Welcome to Paciloka!=====
```

```
Masukkan perintah : 2  
Masukkan nama hotel : Wisma Gaung  
Hotel dengan nama Wisma Gaung mempunyai profit sebesar 0
```

```
=====Welcome to Paciloka!=====
```

```
Masukkan perintah : 3  
Masukkan nama user : Asep  
Nama user tidak ditemukan di sistem!
```

```
=====Welcome to Paciloka!=====
```

```
Masukkan perintah : 3  
Masukkan nama user : Dek depe  
User dengan nama Dek Depe mempunyai saldo sebesar 2000000
```

```
=====Welcome to Paciloka!=====
```

```
Masukkan perintah : 4  
Masukkan nama user : Asep  
Nama user tidak ditemukan di sistem!
```

```
=====Welcome to Paciloka!=====
```

```
Masukkan perintah : 4  
Masukkan nama user : Dek depe  
Masukkan jumlah uang yang akan ditambahkan ke user : -500000  
Jumlah saldo yang akan ditambahkan ke user harus lebih dari 0!
```

```
=====Welcome to Paciloka!=====
```

```
Masukkan perintah : 4
Masukkan nama user : Dek depe
Masukkan jumlah uang yang akan ditambahkan ke user : 500000
Berhasil menambahkan 500000 ke user Dek depe. Saldo user menjadi
2500000

Masukkan perintah : 8
Terima kasih sudah mengunjungi Paciloka!
```

Test Case 3

```
Masukan banyak hotel : 3
Masukan banyak user : 2

Masukan nama hotel ke-1 : Hotel A
Masukan banyak kamar hotel ke-1 : 1
Masukan harga satu kamar hotel ke-1 : 10000
Masukan nama hotel ke-2 : Hotel B
Masukan banyak kamar hotel ke-2 : 2
Masukan harga satu kamar hotel ke-2 : 10000
Masukan nama hotel ke-3 : Hotel C
Masukan banyak kamar hotel ke-3 : 3
Masukan harga satu kamar hotel ke-3 : 10000

Masukan nama user ke-1 : Pak Cil
Masukan saldo user ke-1 : 20000
Masukan nama user ke-2 : Bu Cil
Masukan saldo user ke-2 : 20000

=====Welcome to Paciloka=====

Masukkan perintah : 6
Masukkan nama hotel : Hotel A
Hotel Hotel A tidak memiliki pelanggan!

=====Welcome to Paciloka=====

Masukkan perintah : 7
Masukkan nama user : Pak Cil
User Pak Cil tidak pernah melakukan booking!

=====Welcome to Paciloka=====

Masukkan perintah : 5
```

```
Masukkan nama user : Pak C
Nama user tidak ditemukan di sistem!

=====Welcome to Paciloka=====

Masukkan perintah : 5
Masukkan nama user : Pak Cil
Masukkan nama hotel : Argonda
Nama hotel tidak ditemukan di sistem!

=====Welcome to Paciloka=====

Masukkan perintah : 5
Masukkan nama user : Pak Cil
Masukkan nama hotel : Hotel A
Masukkan jumlah kamar yang akan di-booking : -2
Jumlah kamar yang akan dipesan harus lebih dari 0!

=====Welcome to Paciloka=====

Masukkan perintah : 5
Masukkan nama user : Pak Cil
Masukkan nama hotel : Hotel A
Masukkan jumlah kamar yang akan di-booking : 10
Booking tidak berhasil!

=====Welcome to Paciloka=====

Masukkan perintah : 5
Masukkan nama user : Pak Cil
Masukkan nama hotel : Hotel A
Masukkan jumlah kamar yang akan di-booking : 1
User dengan nama Pak Cil berhasil melakukan booking di hotel Hotel A
dengan jumlah 1 kamar!

=====Welcome to Paciloka=====

Masukkan perintah : 6
Masukkan nama hotel : Hotel A
Hotel A | Pak Cil

=====Welcome to Paciloka=====

Masukkan perintah : 5
Masukkan nama user : Bu Cil
```

```
Masukkan nama hotel : Hotel A
Masukkan jumlah kamar yang akan di-booking : 1
Booking tidak berhasil!

=====Welcome to Paciloka=====

Masukkan perintah : 5
Masukkan nama user : Pak Cil
Masukkan nama hotel : Hotel B
Masukkan jumlah kamar yang akan di-booking : 1
User dengan nama Pak Cil berhasil melakukan booking di hotel Hotel B
dengan jumlah 1 kamar!

=====Welcome to Paciloka=====

Masukkan perintah : 5
Masukkan nama user : Bu Cil
Masukkan nama hotel : Hotel B
Masukkan jumlah kamar yang akan di-booking : 1
User dengan nama Bu Cil berhasil melakukan booking di hotel Hotel B
dengan jumlah 1 kamar!

=====Welcome to Paciloka=====

Masukkan perintah : 6
Masukkan nama hotel : Hotel B
Hotel B | Pak Cil, Bu Cil

=====Welcome to Paciloka=====

Masukkan perintah : 7
Masukkan nama user : Pak Cil
Pak Cil | Hotel A, Hotel B

=====Welcome to Paciloka=====

Masukkan perintah : 7
Masukkan nama user : Bu Cil
Bu Cil | Hotel B

=====Welcome to Paciloka=====

Masukkan perintah : 8
Terima kasih sudah mengunjungi Paciloka!
```

Komponen Penilaian

- **40%** Kebenaran fungsionalitas program
- **35%** Kebenaran implementasi OOP
- **10%** Memenuhi kriteria standar penulisan kode Python*
- **10%** Mengumpulkan dengan format dan penamaan file yang benar
- **5%** Dokumentasi kode

*Standar penulisan kode yang harus dipenuhi yaitu:

1. Indentasi yang konsisten
2. Aturan penamaan variabel mengikuti Python Naming Convention ([sumber](#))
3. Penamaan Module, Class, Method, dan Variabel yang tidak ambigu

Berkas yang Perlu Dikumpulkan

- paciloka.py

Kumpulkan berkas paciloka.py dalam sebuah file .zip dengan format penamaan seperti berikut:

[KodeAsdos]_[Kelas]_[NPM]_[NamaLengkap]_Lab08.zip

Contoh:

FWS_F_2006596535_FranciscoWilliamSudianto_Lab08.zip