

## **Lab 09**

### **OOP Advanced**



FAKULTAS  
**ILMU**  
**KOMPUTER**

---

## **Paciloka HRM System**

Sub-CPMK 10 : Mampu menjelaskan fitur kompleks pada object-oriented programming



source:

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

Setelah Pak Esde berhasil membuat Paciloka, ternyata Paciloka membawa keuntungan yang sangat besar. Maka dari itu, Pak Esde berpikir untuk mengembangkan bisnisnya di sektor perhotelan. Ia ingin membuat bisnis hotelnya sendiri. Untuk itu, ia membutuhkan program untuk mengelola kinerja dari para pekerjanya. Pak Esde kembali meminta Dek Depe untuk membuat programnya, tetapi melihat Dek Depe yang kewalahan dengan permintaan Pak Esde, kamu sebagai teman yang baik pun berniat menolongnya.

## Class Person

Person merupakan semua orang yang bekerja di hotel tersebut. Person memiliki atribut sebagai berikut.

- Name: nama dari Person
- Payment: jumlah bayaran untuk tiap pekerjaan yang dikerjakan
- Stamina: nilai yang menentukan worker mampu bekerja atau tidak. Jika nilainya  $\leq 0$ , maka Person tidak bisa bekerja
- Total\_work: jumlah pekerjaan yang dilakukan

Semua atribut memiliki sifat *private*. Stamina dan total\_work memiliki *setter*, sedangkan name dan payment tidak memiliki *setter*. Semua atribut memiliki *getter*.

Setiap Person memiliki method berikut:

- pay\_day(): menghitung berapa banyak gaji yang diperoleh berdasarkan rumus  $\text{payment} * \text{total\_work}$ .
- is\_available(cost\_stamina): mengecek apakah Worker mampu mengerjakan pekerjaan tersebut berdasarkan jumlah stamina. Worker hanya bisa mengerjakan pekerjaan tersebut jika stamina yang dimiliki  $\geq$  stamina yang dibutuhkan untuk mengerjakan pekerjaan itu.
- work(cost\_stamina): method untuk bekerja dan mengurangi stamina berdasarkan cost\_stamina dari pekerjaan tersebut. Method ini juga menambahkan total\_work sebanyak 1 setiap kali dipanggil.

## Class Worker

Worker adalah subclass dari Person, tetapi atribut **name pada Worker harus unik**.

Saat pertama kali dibuat, atribut **stamina** pada Worker bernilai **100** dan atribut **payment** bernilai **5000**. Worker memiliki atribut tambahan sebagai berikut:

- Bonus: akumulasi bonus yang diterima dari mengerjakan berbagai pekerjaan. Atribut ini bersifat *private* dan memiliki *setter getter*.

Method tambahan yang ada di Worker adalah sebagai berikut:

- work(bonus, cost\_stamina): method untuk bekerja (jangan lupa untuk memastikan stamina yang dibutuhkan cukup) dan mengurangi stamina berdasarkan cost\_stamina. Selain itu, juga menambahkan bonus ke atribut akumulasi bonus. Method ini juga menambahkan total\_work sebanyak 1 setiap kali dipanggil.

## Class Manager

Manager merupakan subclass dari Person. Manager bisa memberikan pekerjaan bagi Worker, merekrut Worker baru, dan memecat Worker yang ada. Atribut baru yang dimiliki Manager adalah:

- **List\_worker:** merupakan list dari semua Worker yang masih bekerja disitu. Atribut ini bersifat private dan memiliki getter.

Method baru yang hanya dimiliki Manager adalah:

- **Hire\_worker(name):** berguna untuk merekrut Worker baru dan menambahkannya ke list\_worker. Worker baru tidak bisa direkrut jika namanya sama dengan worker yang sudah bekerja disitu (Input nama tidak dijamin unik).
- **Fire\_worker(name):** berguna untuk memecat Worker dan mengeluarkannya dari list\_worker. Jika nama yang dimasukkan tidak ada dalam list\_worker, maka program akan mencetak “Nama tidak ditemukan”
- **Give\_work(name, bonus, cost\_stamina):** memberikan pekerjaan kepada worker dan menambahkan akumulasi bonus pada worker, serta mengurangi stamina worker sesuai cost\_stamina.

Semua method diatas akan mengurangi stamina Manager sebesar 10 dan menambahkan atribut total\_work sebanyak 1.



Untuk mempermudah penggeraan, kamu dapat menggunakan [template](#) yang sudah disediakan. Implementasi tidak wajib menggunakan template, asalkan outputnya benar.

## Ketentuan Program

### Alur Program

Di awal, program akan meminta input berupa :

1. **Nama manajer (String)** : nama manajer yang ada di sistem
2. **Jumlah pembayaran (int)** : jumlah pembayaran manajer (gaji)
3. **Stamina manajer (int)** : stamina awal manajer

Lalu, program akan terus meminta input perintah sebagai berikut :

- **Daftar Perintah**

1. **Lihat status pegawai**

Perintah ini **diwakili dengan angka 1**. Mahasiswa diminta untuk mencetak seluruh nama pegawai (termasuk manajer) yang ada di dalam sistem dengan format yang sudah ditentukan. **Hasil pencetakan tidak harus terurut.**

2. **Beri tugas**

Perintah ini **diwakili dengan angka 2**. Mahasiswa diminta untuk memberikan tugas ke pegawai <nama\_pegawai> beserta <bonus\_pekerjaan> dan <beban\_stamina> jika melakukan tugas tersebut. Input <nama\_pegawai> tidak dijamin ada di dalam sistem.

Jika <nama\_pegawai> ditemukan dan/atau memiliki stamina yang cukup, cetak output yang sesuai dengan format yang sudah ditentukan. Jika <nama\_pegawai> tidak ditemukan dan/atau tidak memiliki stamina yang cukup, cetak pesan error sesuai dengan format yang sudah ditentukan.

3. **Cari pegawai baru**

Perintah ini **diwakili dengan angka 3**. Mahasiswa diminta untuk menambahkan pegawai baru dengan <nama\_pegawai>. Berikan output yang sesuai dengan yang sudah dijelaskan pada method `Hire_worker`.

4. **Pecat pegawai**

Perintah ini **diwakili dengan angka 4**. Mahasiswa diminta untuk memecat pegawai <nama\_pegawai> dari sistem dan menghapusnya dari daftar status pegawai.

5. **Keluar**

Perintah ini **diwakili dengan angka 5**. 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**

## Test Case



Teks berwarna merah adalah masukan dari mahasiswa

### Test case 1:

```
Masukkan nama manajer: Bambang  
Masukkan jumlah pembayaran: 10000  
Masukkan stamina manajer: 100
```

PACILOKA

- ```
-----  
1. Lihat status pegawai  
2. Beri tugas  
3. Cari pegawai baru  
4. Pecat pegawai  
5. Keluar
```
- 

```
Masukkan pilihan: 1  
Manager | Bambang | Total kerja:  
0 | Stamina: 100 | Gaji:  
0
```

PACILOKA

- ```
-----  
1. Lihat status pegawai  
2. Beri tugas  
3. Cari pegawai baru  
4. Pecat pegawai  
5. Keluar
```
- 

```
Masukkan pilihan: 3  
Nama pegawai baru: Udin  
Berhasil mendapat pegawai baru
```

PACILOKA

- ```
-----  
1. Lihat status pegawai  
2. Beri tugas  
3. Cari pegawai baru  
4. Pecat pegawai  
5. Keluar
```
-

```
Masukkan pilihan: 1
Worker | Udin | Total kerja:
0 | Stamina: 100 | Gaji:
0
Manager | Bambang | Total kerja:
1 | Stamina: 90 | Gaji:
10000
```

PACILOKA

- 
- 1. Lihat status pegawai
  - 2. Beri tugas
  - 3. Cari pegawai baru
  - 4. Pecat pegawai
  - 5. Keluar
- 

```
Masukkan pilihan: 2
Tugas akan diberikan kepada: Udin
Bonus pekerjaan: 1000
Beban stamina: 20
Hasil cek ketersediaan pegawai:
Pegawai dapat menerima pekerjaan
=====
Berhasil memberi pekerjaan kepada Udin
```

PACILOKA

- 
- 1. Lihat status pegawai
  - 2. Beri tugas
  - 3. Cari pegawai baru
  - 4. Pecat pegawai
  - 5. Keluar
- 

```
Masukkan pilihan: 1
Worker | Udin | Total kerja:
1 | Stamina: 80 | Gaji:
6000
Manager | Bambang | Total kerja:
2 | Stamina: 80 | Gaji:
20000
```

PACILOKA

1. Lihat status pegawai
  2. Beri tugas
  3. Cari pegawai baru
  4. Pecat pegawai
  5. Keluar
- 

Masukkan pilihan: 5

---

Berhenti mengawasi hotel, sampai jumpa !

---

### Test case 2:

Masukkan nama manajer: **Bambang**  
Masukkan jumlah pembayaran: **100000**  
Masukkan stamina manajer: **100**

PACILOKA

---

1. Lihat status pegawai
  2. Beri tugas
  3. Cari pegawai baru
  4. Pecat pegawai
  5. Keluar
- 

Masukkan pilihan: 3

Nama pegawai baru: **Udin**

**Berhasil mendapat pegawai baru**

PACILOKA

---

1. Lihat status pegawai
  2. Beri tugas
  3. Cari pegawai baru
  4. Pecat pegawai
  5. Keluar
- 

Masukkan pilihan: 3

Nama pegawai baru: **udin**

**Sudah ada!**

PACILOKA

- 
- 1. Lihat status pegawai
  - 2. Beri tugas
  - 3. Cari pegawai baru
  - 4. Pecat pegawai
  - 5. Keluar
- 

Masukkan pilihan: 2

Tugas akan diberikan kepada: Udin

Bonus pekerjaan: 20000

Beban stamina: 50

Hasil cek ketersediaan pegawai:

Pegawai dapat menerima pekerjaan

=====

Berhasil memberi pekerjaan kepada Udin

PACILOKA

- 
- 1. Lihat status pegawai
  - 2. Beri tugas
  - 3. Cari pegawai baru
  - 4. Pecat pegawai
  - 5. Keluar
- 

Masukkan pilihan: 2

Tugas akan diberikan kepada: Udin

Bonus pekerjaan: 30000

Beban stamina: 50

Hasil cek ketersediaan pegawai:

Pegawai dapat menerima pekerjaan

=====

Berhasil memberi pekerjaan kepada Udin

PACILOKA

- 
- 1. Lihat status pegawai
  - 2. Beri tugas
  - 3. Cari pegawai baru
  - 4. Pecat pegawai
  - 5. Keluar
- 

Masukkan pilihan: 2

Tugas akan diberikan kepada: **Udin**

Bonus pekerjaan: **10000**

Beban stamina: **10**

**Hasil cek ketersediaan pegawai:**

Pegawai tidak dapat menerima pekerjaan. Stamina pegawai tidak cukup.

PACILOKA

- 
- 1. Lihat status pegawai
  - 2. Beri tugas
  - 3. Cari pegawai baru
  - 4. Pecat pegawai
  - 5. Keluar
- 

Masukkan pilihan: **4**

Nama pegawai yang akan dipecat: **Andi**

**Nama tidak ditemukan**

PACILOKA

- 
- 1. Lihat status pegawai
  - 2. Beri tugas
  - 3. Cari pegawai baru
  - 4. Pecat pegawai
  - 5. Keluar
- 

Masukkan pilihan: **4**

Nama pegawai yang akan dipecat: **Udin**

**Berhasil memecat Udin**

PACILOKA

- 
- 1. Lihat status pegawai
  - 2. Beri tugas
  - 3. Cari pegawai baru
  - 4. Pecat pegawai
  - 5. Keluar
- 

Masukkan pilihan: **5**

---

**Berhenti mengawasi hotel, sampai jumpa !**

**Test case 3:**

```
Masukkan nama manajer: Bambang  
Masukkan jumlah pembayaran: 100000  
Masukkan stamina manajer: 20
```

```
PACILOKA
```

- ```
-----  
1. Lihat status pegawai  
2. Beri tugas  
3. Cari pegawai baru  
4. Pecat pegawai  
5. Keluar
```
- 

```
Masukkan pilihan: 3  
Nama pegawai baru: Udin  
Berhasil mendapat pegawai baru
```

```
PACILOKA
```

- ```
-----  
1. Lihat status pegawai  
2. Beri tugas  
3. Cari pegawai baru  
4. Pecat pegawai  
5. Keluar
```
- 

```
Masukkan pilihan: 2  
Tugas akan diberikan kepada: Udin  
Bonus pekerjaan: 10000  
Beban stamina: 10  
Hasil cek ketersediaan pegawai:  
Pegawai dapat menerima pekerjaan  
=====  
Berhasil memberi pekerjaan kepada Udin  
  
-----  
Stamina manajer sudah habis, sampai jumpa !  
-----
```

## Komponen Penilaian

- **45%** Kebenaran fungsionalitas program
- **30%** Menggunakan **inheritance** dan **polymorphism**
- **5%** Dokumentasi kode
- **10%** Memenuhi kriteria standar penulisan kode Python\*
- **10%** Mengumpulkan dengan format dan penamaan file yang benar

\*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

- lab09.zip

Kumpulkan berkas `lab09.zip` yang telah di-zip dengan format penamaan seperti berikut.

[KodeAsdos]\_[Kelas]\_[NPM]\_[NamaLengkap]\_Lab09.zip

Contoh:

TYR\_A\_1234567890\_DhinaRotuaMutiara\_Lab09.zip