

LATIHAN

1. Buatlah sebuah kelas **Handphone** yang dapat menyimpan beberapa atribut seperti merk, model, RAM, jumlah stok, dan harga HP tersebut. Lalu, buatlah **3 buah instansi** dari kelas tersebut sehingga menghasilkan 3 model HP yang berbeda. Contoh:

Handphone1: "Samsung", "Galaxy M52", "8 GB", **200**, 5399000

Handphone2: "Xiaomi", "Redmi Note 10", "4 GB", 300, 2499000

Handphone3: "Apple", "iPhone SE 2020", 3 GB, 100, 7999000

Jumlah stok masing-masing HP dapat diubah menggunakan fungsi **setter ubah_stok(stok)** dan informasi stok tersebut dapat dilihat melalui fungsi **getter lihat_stok()** dengan contoh output untuk "Handphone1" seperti berikut:

ubah_stok(**180**): "Stok Samsung Galaxy M52 terjual dari **200** menjadi **180** buah"

lihat_stok(): "Samsung Galaxy M52 (8 GB) dengan harga Rp 5399000 tersisa 180 buah"

2. Buatlah sebuah kelas **Pemain** dengan ketentuan sebagai berikut:
 - Mempunyai atribut kelas berupa `__nama` dan `__health_point` (atribut bersifat **private**; konsep visibilitas secara lebih jelas diajarkan minggu depan). Atribut nama dapat diisi secara bebas oleh pengguna saat instansiasi kelas
 - Nilai mula-mula dari atribut `health_point` adalah 30
 - Terdapat **setter** dan **getter** yang dapat dijalankan untuk mengubah dan melihat nilai `health_point` (gunakan konsep **property decorator**, **tanpa** membuat fungsi bantuan dengan nama lainnya)
 - Buatlah juga property **getter nama** untuk menampilkan nama pemain nantinya
 - Buatlah **2 buah instansiasi** pemain berdasarkan kelas Pemain tersebut, misalnya dengan nama PemainA dan PemainB
 - PemainA dan PemainB dapat saling menyerang dengan menerapkan **perulangan** permainan **gunting batu kertas** (modifikasi kode dari minggu sebelumnya). Kode tersebut diletakkan di luar kelas Pemain (yaitu pada main program)
 - Jika salah satu pemain kalah maka nilai `health_point` nya **dikurangi 10 poin** dengan memanggil fungsi **setter health_point** yang telah dibuat sebelumnya
 - Permainan berakhir ketika `health_point` salah satu pemain bernilai nol, gunakan fungsi **getter** untuk mengecek nilai `health_point` di tiap-tiap perulangan gunting-batu-kertas

Contoh output adalah sebagai berikut (tidak mesti sama persis):

Permainan gunting batu kertas
Pilih 1 untuk batu, 2 untuk kertas, 3 untuk gunting
Nama pemain A: Bambang
Nama pemain B: Naufal

Bambang 30 health - Naufal 30 health
Tangan Bambang: 1
Tangan Naufal: 2
Naufal menang

Bambang 20 health - Naufal 30 health
Tangan Bambang: 3
Tangan Naufal: 2
Bambang menang

Bambang 20 health - Naufal 20 health
Tangan Bambang: 2
Tangan Naufal: 3
Naufal menang

Bambang 10 health - Naufal 20 health
Tangan Bambang: 3
Tangan Naufal: 1
Naufal menang

TUGAS

1. Buatlah sebuah kelas (singleton) **TokoBuku** yang memiliki sebuah atribut berupa **__list_buku** (bersifat private). List buku tersebut nantinya dapat diisi dengan buku-buku yang bertipe data **dictionary**, dengan key yang wajib ada yaitu **nama**, **genre**, **penulis**, dan **tahun** terbit buku. Saat pertama kali kelas diinstansiasi, user akan diminta untuk menambahkan sebanyak X buku terlebih dahulu (gunakan **konstruktor**). Selain menambah buku baru, kelas tersebut juga harus dapat menerapkan fungsi **CRUD** (create, read, update, delete) lainnya untuk memodifikasi list buku **tanpa menggunakan magic method**. Contoh output menu (tidak mesti sama persis):

```
Masukkan jumlah buku mula-mula: 2
# ... (masukkan input data buku sebanyak 2 kali)
```

```
Toko Buku Gr4m3dia
1) Tambah buku baru
2) Lihat list buku
3) Modifikasi buku lama
4) Hapus buku lama
5) Keluar program
Masukkan nomor input: 1
```

```
Masukkan nama buku: Nebula
Masukkan genre buku: Fantasi
Masukkan penulis buku: Tere Liye
Masukkan tahun terbit buku: 2019
Buku berhasil ditambahkan!
```

```
Toko Buku Gr4m3dia
1) Tambah buku baru
2) Lihat list buku
3) Modifikasi buku lama
4) Hapus buku lama
5) Keluar program
Masukkan nomor input: 2
```

```
Belajar PBO    Edukasi    Asprak PBO    2022
Sang Pemimpi   Roman    Andrea Hirata  2006
Nebula         Fantasi   Tere Liye     2019
```

```
# ... (tampilkan ulang tampilan awal)
Masukkan nomor input: 4
```

```
Masukkan judul yang ingin dihapus: Nebula
Buku berhasil dihapus!
```

```
# ... (tampilkan ulang tampilan awal)
Masukkan nomor input: 5
```

2. Buatlah sebuah kelas (singleton) yang dapat mengimplementasikan Vigenere cipher, dimana saat kelas diinstansiasi pertama kali akan meminta sebuah input berupa **key** (kunci). Kemudian buatlah fungsi **enkripsi**, **dekripsi**, dan **ganti key** untuk dapat dieksekusi kemudian. Contoh output adalah sebagai berikut (tidak mesti sama persis):

Masukkan kunci (key): PBO

Vigenere cipher

- 1) Enkripsi plainteks
- 2) Dekripsi cipher
- 3) Ganti kunci (key)
- 4) Keluar program

Masukkan nomor input: 1

Masukkan plainteks: TEKNIK INFORMATIKA ITERA

Hasil enkripsi: IFYCJY XOTDSAPUWZB WIFFP

Vigenere cipher

- 1) Enkripsi plainteks
- 2) Dekripsi cipher
- 3) Ganti kunci (key)
- 4) Keluar program

Masukkan nomor input: 2

Masukkan cipher: ESOZUWZVA ZFRJB

Hasil dekripsi: PRAKTIKUM KEDUA

... (tampilkan ulang tampilan awal)

Masukkan nomor input: 4

Cara kerja enkripsi Vigenere cipher adalah sebagai berikut:

Key (input): PBO

Plainteks: TEKNIK INFORMATIKA ITERA

Ulangi kunci hingga setara dengan panjang plainteks

TEKNIK INFORMATIKA ITERA

PBOPBO PBOPBOPBOPB OPBOP

Lihat indeks huruf alfabet berikut

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Enkripsi huruf per huruf (modulo total alfabet)

$T + P = (19 + 15) \% 26 = 34 \% 26 = 8$ (I)

$E + B = (4 + 1) \% 26 = 5 \% 26 = 5$ (F)

$K + O = (10 + 14) \% 26 = 24 \% 26 = 24$ (Y)

$N + P = (13 + 15) \% 26 = 28 \% 26 = 2$ (C)

$I + B = (8 + 1) \% 26 = 9 \% 26 = 9$ (J)

$K + O = (10 + 14) \% 26 = 24 \% 26 = 24$ (Y)

... dan seterusnya

Hasil akhir

Plainteks: TEKNIK INFORMATIKA ITERA

Hasil enkripsi: IFYCJY XOTDSAPUWZB WIFFP

3. Implementasikanlah sebuah konsep kelas (bebas) yang menurutmu menarik dan ideal, topik dapat berupa apa saja seperti game, dsb. Gunakan berbagai konsep yang telah dipelajari sebelumnya seperti constructor, destructor, property decorator, dan magic method untuk memperluas pemahaman Anda (tidak mesti semua konsep, jika tidak memungkinkan/aneh untuk diterapkan).

Buatlah konsep kelas yang berbeda dengan soal-soal sebelumnya jika memungkinkan, misal jangan membuat lagi kelas manajemen barang dengan konsep yang mirip TokoBuku (soal nomor 1) tanpa tambahan fitur/metode lainnya. Jelaskan latar belakang (atau tujuan), fitur-fitur yang tersedia, dan fungsi kode-kode pada kelas tersebut secara mendetail.

Note: Gunakan konsep kelas dasar; Konsep seperti meta class, polymorphism, dll akan diajarkan minggu-minggu selanjutnya