

## LATIHAN

1. Buatlah sebuah kelas **Mobil** yang berisikan 2 buah atribut terenkapsulasi dengan akses **private**, yaitu **bensin** (0-60 liter) dan **odometer** (dalam kilometer). Mobil tersebut mengkonsumsi 1 liter bensin per kilometer. Kelas tersebut juga memiliki fungsi berupa **isi\_bensin()**, **mengendarai(kilometer)**, dan **lihat\_info()**. Contoh penggunaan kelas yaitu seperti berikut:

```
mobil = Mobil()
mobil.lihat_info()
mobil.isi_bensin()
mobil.lihat_info()
mobil.mengendarai(20)
mobil.lihat_info()
mobil.mengendarai(50)
mobil.lihat_info()
mobil.mengendarai(10)
mobil.lihat_info()
mobil.isi_bensin()
mobil.isi_bensin()
mobil.lihat_info()
```

Hasil output dari kode tersebut adalah:

```
Odometer berada pada angka 0 km, bensin yang tersisa 0 liter
Odometer berada pada angka 0 km, bensin yang tersisa 60 liter
Odometer berada pada angka 20 km, bensin yang tersisa 40 liter
Odometer berada pada angka 60 km, bensin yang tersisa 0 liter
Odometer berada pada angka 60 km, bensin yang tersisa 0 liter
Odometer berada pada angka 60 km, bensin yang tersisa 60 liter
```

2. Buatlah sebuah kelas **Dagangan** yang memiliki atribut instance berupa **nama**, **stok**, dan **harga** (bersifat **private**) yang nilainya tidak sama untuk tiap instansinya. Selain itu, kelas **Dagangan** tersebut juga memiliki atribut kelas berupa **jumlah\_barang** dan **list\_barang** (berisi data-data nama, stok, dan harga barang tiap instansi) yang konsisten pada tiap instansi **Dagangan** yang telah dibuat. Gunakan fungsi bantuan yaitu **lihat\_barang()** untuk menampilkan output dari **jumlah\_barang** dan **list\_barang**. Contoh main program (kalian cukup buat kode untuk class **Dagangan**):

```
Dagangan1 = Dagangan("Galon Aqua 19L", 32, 17000)
Dagangan2 = Dagangan("Gas LPG 5 kg", 22, 88000)
Dagangan3 = Dagangan("Beras Ramos 5 kg", 13, 68000)

Dagangan.lihat_barang()
del Dagangan1
Dagangan.lihat_barang()
```

Output program tersebut (tiap instansi akan memberikan nilai konsisten satu sama lain):

```
Jumlah barang dagangan pada toko: 3 buah
1. Galon Aqua 19L seharga Rp 17000 (stok: 32)
2. Gas LPG 5 kg seharga Rp 88000 (stok: 22)
3. Beras Ramos 5 kg seharga Rp 68000 (stok: 13)
```

Galon Aqua 19L dihapus dari toko!

Jumlah barang dagangan pada toko: 2 buah

1. Gas LPG 5 kg seharga Rp 88000 (stok: 22)
2. Beras Ramos 5 kg seharga Rp 68000 (stok: 13)

Gas LPG 5 kg dihapus dari toko!

Beras Ramos 5 kg dihapus dari toko!

## TUGAS

1. Buatlah sebuah game Minesweeper seperti output di bawah ini, dimana data tiap-tiap kotak (area) disimpan dalam kelas **Kotak** (dalam kasus di bawah, terdapat 9 kelas Kotak yang diinstansiasi).

Kelas Kotak ini mempunyai atribut **instance private** berupa **isi** (bom/kosong), **status** (belum dibuka/sudah dibuka); atribut **kelas publik** berupa **jumlah\_bom** (nilainya konsisten antar instansi tiap objek kotak); serta **fungsi publik** berupa **tampilkan** ("?" jika belum dibuka, "x" jika bom, dan "o" jika kosong) dan **buka\_kotak** untuk mengubah status kotak dari belum dibuka menjadi sudah dibuka. Atribut isi kotak (bom/kosong) bersifat acak dan ditentukan saat pertama kali kotak diinisiasi (gunakan import random bila dibutuhkan).

Implementasi kelas/fungsi bantuan lainnya diperbolehkan dan sangat dianjurkan, karena game tidak akan berjalan jika hanya memanfaatkan kelas Kotak. Output program tidak mesti sama persis seperti contoh di bawah.

```
Permainan Minesweeper
Masukkan dimensi area: 3
```

```
? ? ?
? ? ?
? ? ?
```

```
Masukkan nomor kotak yang ingin dibuka (1-9): 6
```

```
? ? ?
? ? o
? ? ?
```

```
Selamat! Kotak tersebut tidak berisi bom.
```

```
Masukkan nomor kotak yang ingin dibuka (1-9): 8
```

```
? ? ?
? ? o
? o ?
```

```
Selamat! Kotak tersebut tidak berisi bom.
```

```
Masukkan nomor kotak yang ingin dibuka (1-9): 1
```

```
x ? ?
? ? o
? o ?
```

```
Game over! Kotak tersebut berisi bom.
```

```
# Jika menang dan jumlah_bom habis: Selamat! Anda telah memenangkan game.
```

**Catatan:** ide dari soal ini adalah program/seseorang tidak akan tahu apa isi dari suatu kotak sampai kotak tersebut dibuka. Contoh tersebut adalah penerapan konsep abstraksi.

2. Buatlah kelas **AkunBank** yang memiliki atribut instance berupa **no\_pelanggan**, **nama\_pelanggan**, dan **jumlah\_saldo** (bersifat private). Kelas AkunBank tersebut juga memiliki atribut kelas berupa **list\_pelanggan** (berisi data-data pelanggan tiap instansi). Pada kelas AkunBank, terdapat fungsi yang dapat dieksekusi oleh pelanggan, yaitu **lihat\_menu()**,

**lihat\_saldo()**, **tarik\_tunai()**, dan **transfer()** (implementasikan sendiri). Lalu, buatlah 3 instansi AkunBank seperti berikut ini:

```
Akun1 = AkunBank(1234, isi_nama_kalian, 5 000 000 000)
Akun2 = AkunBank(2345, "Ukraina", 6 666 666 666)
Akun3 = AkunBank(3456, "Elon Musk", 9 999 999 999)
```

Melalui Akun1 (akun kalian), simulasikan fungsi-fungsi lihat\_menu(), lihat\_saldo(), dll yang telah disebutkan sebelumnya sehingga menghasilkan output seperti berikut (tidak mesti sama persis, kreasikan sendiri):

```
Selamat datang di Bank Jago
Halo nama_kalian, ingin melakukan apa?
1. Lihat saldo
2. Tarik tunai
3. Transfer saldo
4. Keluar
Masukkan nomor input: 2

Masukkan jumlah nominal yang ingin ditarik: 999999999
Nominal saldo yang Anda punya tidak cukup!
Masukkan jumlah nominal yang ingin ditarik: 200000
Saldo berhasil ditarik!

# Tampilkan ulang menu
Masukkan nomor input: 1

nama_kalian memiliki saldo Rp 4800000

# Tampilkan ulang menu
Masukkan nomor input: 3

Masukkan nominal yang ingin ditransfer: 500000
Masukkan no rekening tujuan: 4321
No rekening tujuan tidak dikenal! Kembali ke menu utama...

# Tampilkan ulang menu
Masukkan nomor input: 3

Masukkan nominal yang ingin ditransfer: 500000
Masukkan no rekening tujuan: 2345
Transfer Rp 500000 ke Ukraina sukses!

# Tampilkan ulang menu
Masukkan nomor input: 4
```

3. Buatlah demonstrasi kelas (topik dan konsep bebas) yang dapat memperlihatkan perbedaan antara atribut/fungsi private, protected, dan public secara jelas (kapan atribut private harus digunakan, kapan atribut public harus digunakan, dsb). Gunakan juga konsep atribut/fungsi kelas (yaitu atribut/fungsi yang bernilai konsisten antar instance) bila memungkinkan untuk mendukung demonstrasi yang Anda buat. Jangan lupa juga untuk memberikan penjelasan/komentar pada baris-baris kode nantinya!