

### No. 1

Buatlah *Class Rectangle* dengan memiliki *properties length* (panjang) dan *width* (lebar) yang bertipe data Double. Memiliki 4 *Constructors*, memiliki 4 buah constructors. Constructor pertama zero parameter, digunakan mengeset *length* dan *width* sama dengan 0.0. Constructor kedua, menerima 1 parameter dengan tipe data double, yaitu *length* dan *width* nilainya adalah sama. Constructor ketiga, menerima 1 parameter dengan tipe data String, yaitu *length* dan *width* nilainya adalah sama, namun dilakukan parsing terlebih dahulu dari tipe data String ke Double. Constructor ke-empat menerima 2 buah parameter. Terdapat 2 Methods dalam kelas ini, yaitu keliling dan luas. Methods yang dimiliki tidak memiliki parameter.

Buat juga *Class RectangleTest* yang memiliki main method. Lakukan instantiasi 4 buah object dengan mencobakan keempat constructors yang berbeda. Lalu, untuk semua objek dilakukan pemanggilan terhadap method keliling dan luas yang nilainya ditampilkan. Dan, ubahlah *length* pada object pertama, dan lakukan lagi perhitungan keliling dan luas untuk objek pertama tersebut!

### No. 2

Pecahan
- pembilang : int
- penyebut : int
+ Pecahan (int pembilang, int penyebut)
+ getPembilang (): int
+ setPembilang (int): void
+ getPenyebut (): int
+ setPenyebut (int): void
+ tambah (Pecahan p): void

Diketahui sebuah class seperti di atas. Konstruktor memiliki 2 paramter untuk menge-set kedua atribut dari class Pecahan. Method tambah merupakan method untuk menambahkan sebuah objek Pecahan dengan objek Pecahan lainnya. Method ini merupakan function dengan nilai yang dikembalikan merupakan hasil penambahan kedua objek Pecahan dengan tipe data class Pecahan tersebut.

Berdasarkan informasi tersebut,

1. Kodekanlah class di samping beserta logika dari setiap method
2. Bentuklah sebuah Main class yang membentuk 2 objek sebagai berikut:

Objek 1: 3/2
Objek 2: 5/6

Setelah membentuk kedua objek, tambahkan kedua objek tersebut, dan tampilkan hasil penambahan kedua objek tersebut. Contoh tampilan program:

$3/2 + 5/6 = 28/12$
---------------------

**No. 3**

Seorang developer ingin membuat sebuah simulasi fighting game menggunakan konsep PBO. Setelah menganalisis kebutuhan, class diagram yang dihasilkan adalah sebagai berikut:

GameOop
- name : String - lifePoint : int - powerTendangan : int - powerPukulan : int
+ GameOop (String name, int PowerTendangan, int PowerPukulan): void + tendang (GameOop karB): void + pukul (GameOop karB): void + getName (): String + setName (String name): void + getLifePoint (): int

Keterangan:

1. Atribut name merupakan identitas karakter
2. Atribut lifePoint merupakan atribut yang merepresentasikan tenaga dari tiap karakter. Nilai default dari lifePoint adalah 100.
3. Atribut powerTendangan merupakan kekuatan yang diberikan ketika melakukan tendangan.
4. Atribut powerPukulan merupakan kekuatan yang diberikan ketika melakukan pukulan.
5. Method “tendang” merupakan method untuk menendang lawan. Masukan dari method ini adalah objek GameCharacter lain. Efek dari method ini mengurangi lifePoint lawan sebanyak powerTendangan yang dimiliki.
6. Method “pukul” merupakan method untuk memukul lawan. Masukan dari method ini adalah objek GameCharacter lain. Efek dari method ini mengurangi lifePoint lawan sebanyak powerPukulan yang dimiliki.
7. Konstruktor digunakan untuk menge-set nama, PowerTendangan, dan PowerPukulan sesuai parameter masukan. Selain itu, konstruktor juga melakukan set lifePoint sebesar nilai default.
8. Method getLifePoint dan getName merupakan method get bagi kedua atribut.
9. Tidak terdapat konstruktor kosong.

Berdasarkan skenario di atas, bantulah developer untuk mengkodekan class di atas. Lalu, pada class Main, berikan scenario berikut:

1. Buat sebuah objek dengan name: “wiro\_sableng”, powerPukulan: 10, powerTendangan: 20.

2. Buat sebuah objek dengan name: “gundala”, powerPukulan: 5, powerTendangan: 25
3. Objek wiro\_sableng memulai pertarungan dengan melakukan tendangan pada objek gundala.
4. Objek gundala melakukan perlawanan dengan menendang balik objek wiro\_sableng.
5. Objek gundala menyerang wiro\_sableng dengan pukulan berturut-turut sebanyak 3x (gunakan perulangan).
6. Pertarungan diakhiri oleh wiro\_sableng dengan melakukan tendangan beruntun 4x pada objek gundala (gunakan perulangan).
7. Tampilkan lifePoint dari objek wiro\_sableng dan objek gundala. Bandingkan kedua lifePoint tersebut.
8. Tampilkan status kemenangan dari pertarungan kedua objek. Objek yang menang adalah objek yang memiliki lifePoint tertinggi di akhir pertarungan.

#### No. 4

Buatlah kelas dengan nama MethodOverload, dimana kelas tersebut memiliki 2 method overload yaitu kuadrat(). Method kuadrat() yang pertama menerima parameter dengan tipe data integer sedangkan method kuadrat() lainnya menerima parameter dengan tipe data double. Kedua return tipe method tersebut void. Buat juga Kelas yang berisi method main dengan nama kelas MethodOverloadTest.

Berikut ini output program setelah dijalankan:

##### DEMO METHOD OVERLOADING

-----

Method 'kuadrat' pertama

Hasil kuadrat 6 adalah: 36

Method 'kuadrat' kedua

Hasil kuadrat 6.45 adalah: 41.6025