

SOAL UGD TIPE C OBJECT ORIENTED TESTING

PT JayaGrup membutuhkan kamu sebagai programmer handal untuk menguji coba method-method yang ada supaya berjalan dengan lancar. Anda diberikan class diagram sebagai berikut.

Kurir
- nama: String - id_kurir : String - gaji_pokok: Double
+ Kurir(String nama, String id_kurir, double gaji_pokok) + getGajiPokok(): Double + hitungTotalGaji(double Tunjangan): void + hitungPajak(double presentase): double + hitungBonus(doubel jumlahPaket, double biayaPerPaket): double + tampil(): string

Anda diminta untuk menguji coba konstruktor dan setiap method yang ada, kecuali method `getGajiPokok()`. Adapun ketentuan pada tiap-tiap konstruktor dan method sebagai berikut:

1. `Kurir(String nama, String id_Kurir, int double gaji_pokok)`
 - Nama tidak boleh kosong, jika salah akan exception.
 - Id_kurir harus 6 digit dan diawali dengan KUXXXX (Contoh KU1234), jika salah akan exception.
 - Gaji_pokok tidak boleh kurang dari 0, jika salah akan exception.
 - Digunakan sebagai konstruktor untuk membuat objek.
2. `hitungTotalGaji(double tunjangan): void`
 - tunjangan tidak boleh kurang dari 0, jika salah akan exception.
 - Rumus `hitungTotalGaji` total gaji = gaji pokok + Bonus + tunjangan – pajak.
 - Untuk parameter pada `hitungBonus` dibebaskan.
 - Untuk double persentase pada `hitungPajak` diisi gabungan satu digit terakhir NPM praktikan.

Contoh 1.

NPM 1: 210711111 **NPM 2:** 210723233

double presentase : 13 atau 31

Contoh 2.

NPM 1: 210711111 **NPM 2:** 210722222 **NPM 3:** 210733333

double presentase : 13 / 31 / 12 / 21 / 23 / 32 (pilih salah satu angka).

- Sertakan komen NPM praktikan pada method `hitungTotalGaji` di kode program.

```
public void hitungTotalGaji(double tunjangan) {  
    //NPM 1 (210711111)  
    //NPM 2 (210722222)  
    //NPM 3 (210733333)
```

3. `hitungPajak(double persentase): double`
 - Persentase tidak boleh kurang dari 0, jika salah akan exception.

- Rumus hitungPajak pajak = gaji_pokok * persentase/100.
- 4. hitungBonus(double jumlahPaket, double biayaPerPaket): double
 - jumlahPaket tidak boleh kurang dari 0, jika salah akan exception.
 - biayaPerPaket tidak boleh kurang dari 0, jika salah akan exception.
 - Rumus hitungBonus bonus = jumlahPaket * biayaPerPaket*10%.
- 5. tampil(): string
 - mengembalikan hasil :
“Kurir dengan (nama) id kurir (id_Kurir) memiliki Gaji (gaji_pokok).”

Ketentuan Soal:

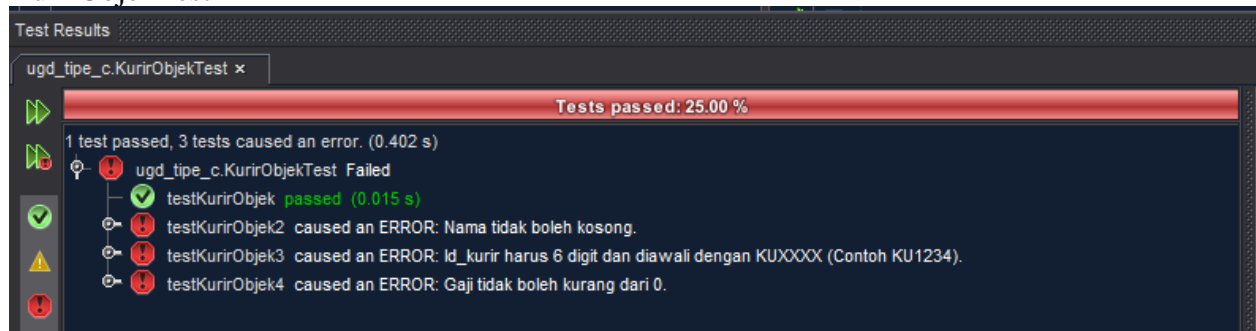
Test class yang dibuat sebanyak 3, yaitu KurirObjekTest, KurirMethodTest, dan KurirTest.

1. Pada **KurirObjekTest** akan menguji pembuatan objek menggunakan konstruktor **Kurir**. Pengujian dilakukan sebanyak 4 kali (4 test case). Setiap test case menggunakan **assertNotNull** dalam pengujian objeknya.
 - a. Test case pertama akan menguji pembuatan objek yang semua nilai atributnya benar.
 - b. Test case kedua akan menguji pembuatan objek yang nama-nya kosong, tapi nilai atribut lainnya benar.
 - c. Test case ketiga akan menguji pembuatan objek yang id_kurir-nya salah, tapi nilai atribut lainnya benar.
 - d. Test case keempat akan menguji pembuatan objek yang gaji_pokok-nya kurang dari 0, tapi nilai atribut lainnya benar.
2. Pada **KurirMethodTest** akan menguji sebanyak 8 kali (8 test case). Sebelum melakukan pengujian, harus menggunakan method set up untuk membuat sebuah objek **Kurir** terlebih dahulu.
 - a. Test case pertama untuk method **hitungTotalGaji** dengan parameter tunjangan kurang dari 0. Menggunakan **assertFalse** untuk membandingkan gaji_pokok setelah eksekusi method hitungTotalGaji tidak sama dengan hasil ekspektasi yang benar. Hasilnya error.
 - b. Test case kedua untuk method **hitungTotalGaji** dengan parameter tunjangan lebih dari 0. Menggunakan **assertFalse** untuk membandingkan gaji_pokok setelah eksekusi method hitungTotalGaji tidak sama dengan hasil ekspektasi yang benar. Hasilnya passed.
 - c. Test case ketiga untuk method **hitungPajak** dengan parameter persentase kurang dari 0. Menggunakan **assertTrue** untuk membandingkan hasil setelah eksekusi method hitungPajak dengan hasil ekspektasi yang benar. Hasilnya error.
 - d. Test case keempat untuk method **hitungPajak** dengan parameter persentase lebih dari 0. Menggunakan **assertTrue** untuk membandingkan hasil setelah eksekusi method hitungPajak dengan hasil ekspektasi yang benar. Hasilnya passed.
 - e. Test case kelima untuk method **hitungBonus** dengan parameter jumlahPaket lebih dari 0 dan biayaPerPaket kurang dari 0. Menggunakan **assertFalse** untuk membandingkan hasil setelah eksekusi method hitungBonus dengan hasil ekspektasi yang salah. Hasilnya error.
 - f. Test case keenam untuk method **hitungBonus** dengan parameter jumlahPaket kurang dari 0 dan biayaPerPaket lebih dari 0. Menggunakan **assertFalse** untuk membandingkan hasil setelah eksekusi method hitungBonus dengan hasil ekspektasi yang salah. Hasilnya error.

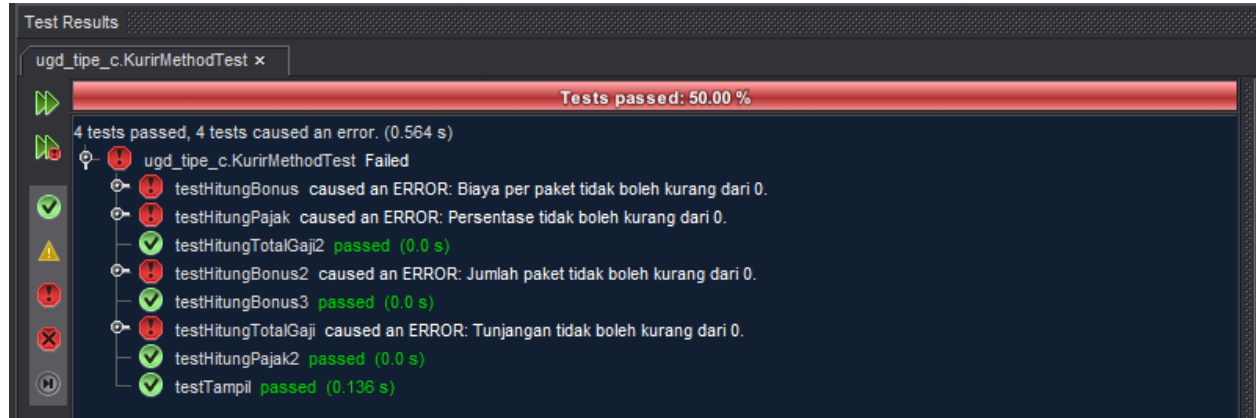
- g. Test case ketujuh untuk method **hitungBonus** dengan parameter jumlahPaket lebih dari 0 dan biayaPerPaket lebih dari 0. Menggunakan **assertEquals** untuk membandingkan hasil setelah eksekusi method hitungBonus dengan hasil ekspektasi yang benar. Hasilnya passed.
 - h. Test case kedelapan untuk method **tampil**. Menggunakan **assertEquals** untuk membandingkan hasil setelah eksekusi method tampil dengan hasil ekspektasi tampil yang benar. Hasilnya passed.
3. Pada KurirTest memanggil unit test KurirObjekTest dan KurirMethodTest. (Hint: Pakai test suite).

Contoh Output:

KurirObjekTest



KurirMethodTest



KurirTest



Ketentuan bonus:

- Selesai UGD dalam waktu 60 menit tanpa tanya dan benar: UGD 110.
- Format pengumpulan bonus: UGD14_X_Y.

SOAL TUGAS

1. Buatlah pre condition, post condition, dan test case dari soal UGD di atas (untuk semua method, kecuali getGajiPokok). Nilai aktual yang digunakan sebagai nilai input juga harus sama persis dengan yang dibuat pada UGD.
2. Jelaskan apa itu object oriented testing menggunakan bahasa kalian masing-masing.

Ketentuan Tugas:

- Selesaikan UGD dengan benar sehingga bisa mengerjakan soal Tugas tanpa ada masalah.
- Kumpulkan penjelasan dalam bentuk docx bukan pdf.
- Format pengumpulan: TGS14_X_Y (X = Kelas, Y = Kelompok).