# Soal Test Java & Java Springboot

# Rules :

Soal diberikan ke kandidat dengan kualifikasi Java / Java Springboot

Kandidat pilih soal 1 / 2 / 3 / 4 / 5

Waktu pengerjaan : 2 jam

Jawaban : zip / github dikirim ke email RTO

RTO kirim ke [juaracodingbootcamp@gmail.com](mailto:juaracodingbootcamp@gmail.com)

Subject email : Test Java – Nomor Soal – nama kandidat

# Soal 1

Buat sebuah program Java untuk mengkonversi detik ke menit. Program meminta pengguna untuk memasukkan jumlah detik yang ingin dikonversi. Contoh:

Input:

Masukkan jumlah detik = 81

Output:

81 detik sama dengan 1 menit 21 detik

# Soal 2

Suatu SPBU menjual produk-produk BBM dengan daftar harga pada tabel berikut:

|  |  |
| --- | --- |
| Produk | Harga per Liter |
| Premium | 6450 |
| Pertalite | 7650 |
| Pertamax | 9000 |

Khusus BBM Premium, karena merupakan BBM subsidi, pembelian maksimal 10 liter.

Buat sebuah program Java, yang meminta pembeli untuk memasukkan jenis BBM yang akan dibeli, kemudian memasukkan jumlah pembelian, kemudian mengeluarkan output jumlah harga yang harus dibayar oleh pembeli. Contoh:

Input:

Masukkan jenis BBM yang ingin dibeli: Pertalite

Masukkan jumlah liter: 55

Output:

Jumlah yang harus dibayar Rp. 420750

Input:

Masukkan jenis BBM yang ingin dibeli: Premium

Masukkan jumlah liter: 5

Output:

Jumlah yang harus dibayar Rp. 32250

Input:

Masukkan jenis BBM yang ingin dibeli: Premium

Masukkan jumlah liter: 15

Output:

Maaf, jumlah pembelian maksimal BBM Premium adalah 10 liter

Jumlah yang harus dibayar Rp. 64500

# Soal 3

Seorang rentenir meminjamkan uang dengan bunga 10% per bulan. Buatlah sebuah program Java yang meminta pengguna untuk memasukkan jumlah uang yang dipinjam dan lama waktu peminjaman (dalam bulan). Program akan menghitung jumlah uang yang harus dibayar peminjam setelah pinjaman jatuh tempo. Contoh:

Input:

Masukkan jumlah pinjaman: 10000000

Masukkan jangka waktu peminjaman (dalam bulan): 4

Output:

Pinjaman yang harus dibayar setelah jatuh tempo: 1464100

Perhitungan:

|  |  |  |  |
| --- | --- | --- | --- |
| Bulan ke | Hutang awal | Bunga (10% \* hutang awal) | Hutang akhir |
| 1 | 1.000.000 | 100.000 | 1.100.000 |
| 2 | 1.100.000 | 110.000 | 1.210.000 |
| 3 | 1.210.000 | 121.000 | 1.331.000 |
| 4 | 1.331.000 | 133.100 | 1.464.100 |

# Soal 4

Sama dengan program nomor sebelumnya. Namun, si rentenir memeriksa pada tiap akhir bulan, apakah si peminjam masih hidup atau sudah meninggal dunia. Jika peminjam sudah meninggal pada akhir bulan, perhitungan bunga hutang dihentikan pada bulan tersebut dan ahli waris cukup membayar hutang berdasarkan perhitungan pada akhir bulan tersebut. Contoh:

Input:

Masukkan jumlah pinjaman: 10000000

Masukkan jangka waktu peminjaman (dalam bulan): 4

Bulan ke 1, apakah peminjam masih hidup? True

Bulan ke 2, apakah peminjam masih hidup? False

Output:

Pinjaman yang harus dibayar oleh ahli waris: 1210000

Input:

Masukkan jumlah pinjaman: 10000000

Masukkan jangka waktu peminjaman (dalam bulan): 4

Bulan ke 1, apakah peminjam masih hidup? False

Output:

Pinjaman yang harus dibayar oleh ahli waris: 1100000

Input:

Masukkan jumlah pinjaman: 10000000

Masukkan jangka waktu peminjaman (dalam bulan): 4

Bulan ke 1, apakah peminjam masih hidup? True

Bulan ke 2, apakah peminjam masih hidup? True

Bulan ke 3, apakah peminjam masih hidup? True

Bulan ke 4, apakah peminjam masih hidup? True

Output:

Pinjaman yang harus dibayar setelah jatuh tempo: 1464100

# Soal 5

Buatlah sebuah class Java untuk menyimulasikan rekening tabungan. Sebuah rekening tabungan memiliki identitas berupa nomor rekening, nama pemilik, dan saldo. Pada rekening tersebut dapat dilakukan proses tarik, setor, dan cek saldo. Proses tarik dan setor akan mengubah saldo, dan menampilkan pesan kepada pengguna bahwa proses tarik atau setor telah berhasil. Pada proses penarikan, perlu dipastikan bahwa jumlah yang ditarik tidak melebihi saldo yang dimiliki. Nama pemilik dan nomor rekening dimasukkan pada saat pertama kali rekening dibuka. Tabungan dapat dibuka dengan setoran awal maupun tanpa setoran awal. Jika rekening dibuka tanpa setoran awal, maka saldo rekening saat dibuka adalah 0.

Main method yang digunakan;

**public** **static** **void** main (String[] args) {

Tabungan simpedes = **new** Tabungan("Tessy Wahyuni Riwayati Hartati", 89, 100000);

simpedes.cekSaldo();

simpedes.setor(8000);

simpedes.cekSaldo();

simpedes.tarik(230000);

simpedes.tarik(60000);

simpedes.cekSaldo();

System.***out***.println();

Tabungan tabanas = **new** Tabungan ("Hindun",90);

tabanas.cekSaldo();

tabanas.tarik(10000);

tabanas.tarik(1000);

tabanas.setor(20000);

tabanas.tarik(20000);

}

Output yang diharapkan;

Saldo anda saat ini adalah 100000

Setoran sebesar 8000 berhasil dilakukan, saldo anda sekarang 108000

Saldo anda saat ini adalah 108000

Penarikan tidak dapat dilakukan karena saldo anda tidak cukup

Penarikan sebesar 60000 berhasil dilakukan, saldo anda sekarang 48000

Saldo anda saat ini adalah 48000

Saldo anda saat ini adalah 0

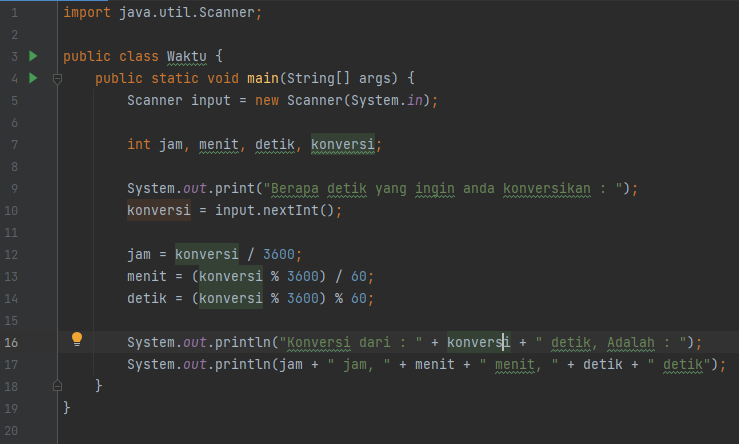
Penarikan tidak dapat dilakukan karena saldo anda tidak cukup

Penarikan tidak dapat dilakukan karena saldo anda tidak cukup

Setoran sebesar 20000 berhasil dilakukan, saldo anda sekarang 20000

Penarikan sebesar 20000 berhasil dilakukan, saldo anda sekarang 0

**Jawab :**

1. 

Output: 