

Can Bismillah
nam . pxtana

Mata Kuliah - Kode Mata Kuliah	Pemrograman Berorientasi Objek-CS4-2023
Hari, tanggal - Kelas	Juni 2024 –CS
Waktu - Sifat	120 Menit – Offline
Dosen Koordinator Mata Kuliah	Randi Farmana Putra. M.Si

Working Instruction:

- Buat code dengan ditulis di kertas yang disediakan dalam bahasa Java
- Ujian bersifat close book
- Berikan komentar pada bagian penting kode untuk menjelaskan apa yang dilakukan oleh bagian tersebut.

Question

(CPL-m, CPMK 1,2 dan 3)

Buatlah sistem manajemen keuangan pribadi yang mencakup berbagai jenis transaksi dan menghitung biaya layanan serta memastikan keabsahan transaksi.

1. Pewarisan

Buat kelas abstrak Transaksi yang memiliki properti dan metode umum untuk semua transaksi.

- Buat kelas abstrak Transaksi dengan properti berikut:


```
public abstrac Transaksi {
    sum .
    ..
}
```

 - String idTransaksi
 - double jumlah
 - String tanggal
- Tambahkan metode abstrak void tampilkanDetail() yang akan diimplementasikan oleh kelas turunan.
- Tambahkan metode konkret void cetakTransaksi() yang mencetak informasi umum transaksi (ID Transaksi: {idTransaksi}, Jumlah: {jumlah}, Tanggal: {tanggal}).
- Buat dua kelas turunan Pembayaran dan Transfer yang mewarisi dari Transaksi.



Segala bentuk kecurangan akademik (termasuk pelanggaran tata tertib ujian) dalam setiap proses pembelajaran akan diberikan sanksi sesuai peraturan yang berlaku



Implementasi:

- implementasikan metode `tampilkanDetail()` yang mencetak ID Transaksi: `{idTransaksi}`, Jumlah: `{jumlah}`, Tanggal: `{tanggal}`.).

2. Interface

Buat dua interface `BiayaLayanan` dan `Verifikasi` dengan metode berikut:

- `BiayaLayanan: double hitungBiaya()`
- `Verifikasi: boolean verifikasiTransaksi()`

a. Implementasikan satu interface pada kelas `Pembayaran` dan dua interface pada kelas `Transfer`.

Implementasi:

- Pada kelas `Pembayaran`, implementasikan metode `hitungBiaya()` yang mengembalikan $1000 + (\text{jumlah} * 0.01)$.
- Pada kelas `Transfer`, implementasikan metode `hitungBiaya()` yang mengembalikan $2500 + (\text{jumlah} * 0.005)$.
- implementasikan metode `verifikasiTransaksi()` yang mengembalikan `true` jika jumlah > 0 dan `false` jika tidak.

b. Tambahkan metode `void konfirmasiPembayaran()` di kelas `Pembayaran` yang mencetak `Pembayaran` dengan ID `{idTransaksi}` telah dikonfirmasi.

c. Tambahkan metode `void konfirmasiTransfer()` di kelas `Transfer` yang mencetak `Transfer` dengan ID `{idTransaksi}` telah dikonfirmasi.

3. Polimorfisme

a. Buat kelas `ManajerKeuangan` yang memiliki metode `prosesTransaksi(Transaksi transaksi)`. Metode ini harus menerima objek dari kelas `Transaksi`, menampilkan detail transaksi, menghitung dan menampilkan biaya layanan, serta memverifikasi transaksi.

Implementasi:

- Gunakan `instanceof` untuk memastikan bahwa metode `hitungBiaya()` dan `verifikasiTransaksi()` hanya dipanggil jika objek yang diteruskan adalah instance dari kelas yang mengimplementasikan interface tersebut.
- Gunakan `instanceof` untuk memanggil metode khusus `konfirmasiPembayaran()` jika objek adalah instance dari `Pembayaran`, dan `konfirmasiTransfer()` jika objek adalah instance dari `Transfer`.

4. Array Objek

- a. Buat array dari objek Transaksi yang mencakup beberapa objek Pembayaran dan Transfer.
- b. Iterasi melalui array dan panggil metode prosesTransaksi untuk setiap transaksi.

5. Kelas Utama

- a. Buat kelas Utama untuk menguji semua fitur di atas.

Nilai Variabel yang Digunakan:

- Pembayaran 1: idTransaksi = "P01", jumlah = 200000, tanggal = "2024-06-01"
- Transfer 1: idTransaksi = "T01", jumlah = 500000, tanggal = "2024-06-02"
- Pembayaran 2: idTransaksi = "P02", jumlah = 150000, tanggal = "2024-06-03"
- Transfer 2: idTransaksi = "T02", jumlah = 750000, tanggal = "2024-06-04"