(Enkapsulasi Pada Pemrograman Berorientasi Objek

## Pertanyaan

* + 1. Pada class TestMobil, saat kita menambah kecepatan untuk pertama kalinya, mengapa

muncul peringatan “Kecepatan tidak bisa bertambah karena Mesin Off!”?

karena pada saat itu, motor masih dalam keadaan mati (kontakOn = false)

* + 1. Mengapat atribut kecepatan dan kontakOn diset private?

Memastikan kelas lain tidak bisa langsung mengakses atribut kecepatan dan kontakOn

* + 1. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

else if (kecepatan >= 100){

            System.out.println("Kecepatan sudah mencapai kecepatan maksimal!\n");

        }

## Pertanyaan – Percobaan 3 dan 4

* + 1. Apa yang dimaksud getter dan setter?

Getter adalah metode untuk mengambil nilai dari atribut dalam sebuah objek yang biasanya bersifat private dan setter digunakan untuk mengubah nilai dari atribut dalam sebuah objek

* + 1. Apa kegunaan dari method getSimpanan()?

getSimpanan() berfungsi untuk mengambil nilai simpanan

* + 1. Method apa yang digunakan untk menambah saldo?

Method setor

* + 1. Apa yand dimaksud konstruktor?

Konstruktor adalah method khusus yang akan dieksekusi pada saat pembuatan objek (instansiasi)

* + 1. Sebutkan aturan dalam membuat konstruktor?

1. Nama konstruktor harus sama dengan nama class

2. Konstruktor tidak memiliki tipe data return

3. Konstruktor tidak boleh menggunakan modifier abstract, static, final, dan syncronized

* + 1. Apakah boleh konstruktor bertipe private?

Ya, kontruktor boleh bertipe private, itu berati konstruktor tersebut hanya bisa dipanggil pada kelas itu sendiri

* + 1. Kapan menggunakan parameter dengan passsing parameter?

Ketika ada atribut yang membutuhkan nilai yang spesifik

* + 1. Apa perbedaan atribut class dan instansiasi atribut?

Atribut class adalah variabel yang dideklarasikan di dalam class sedangkan instansiasi atribut adalah proses pemberian nilai awal kepada atribut class

* + 1. Apa perbedaan class method dan instansiasi method?

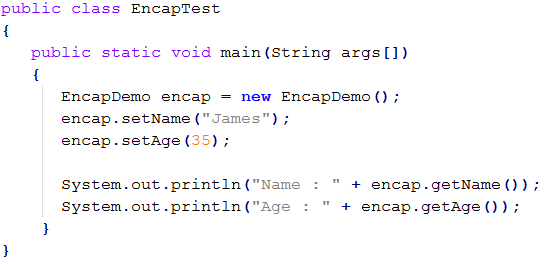
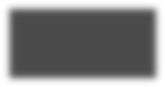
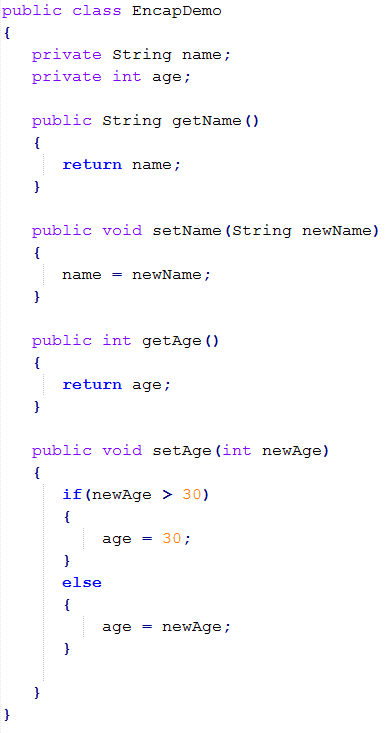
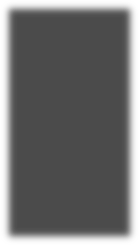
Class method adalah method yang dapat diakses oleh semua objek dari kelas tersebut dan biasanya digunakan untuk melakukan tugas umum dan instansiasi method adalah method yang hanya dapat diakses oleh objek tertentu dan digunakan untuk melakukan tugas yang spesifik

# Kesimpulan

Dari percobaan diatas, telah dipelajari kosep dari enkapsulasi, kontruktor, access modifier yang terdiri dari 4 jenis yaitu public, protected, default dan private. Konsep atribut atau method class yang ada di dalam blok code class dan konsep instansiasi atribut atau method. Cara penggunaan getter dan setter beserta fungsi dari getter dan setter. Dan juga telah dipelajari atau memahami notasi UML

# Tugas

* 1. Cobalah program dibawah ini dan tuliskan hasil outputnya



Output dari program diatas adalah

Name : James

Age : 30

* 1. Pada program diatas, pada class EncapTest kita mengeset age dengan nilai 35, namun pada saat ditampilkan ke layar nilainya 30, jelaskan mengapa.

Karena pada method setAge kita menaruh kondisi jika nilai age yang dimasukkan lebih dari 30, maka nilai age akan menjadi 30

* 1. Ubah program diatas agar atribut age dapat diberi nilai maksimal 30 dan minimal 18.

if(newAge > 30){

            System.out.println("Umur maksimum adalah 30!");

            age = 30;

      } else if(newAge < 18){

            System.out.println("Umur minimum adalah 18!");

            age = 18;

        } else {

            age = newAge;

        }

* 1. Pada sebuah sistem informasi koperasi simpan pinjam, terdapat class Anggota yang memiliki atribut antara lain nomor KTP, nama, limit peminjaman, dan jumlah pinjaman. Anggota dapat meminjam uang dengan batas limit peminjaman yang ditentukan. Anggota juga dapat mengangsur pinjaman. Ketika Anggota tersebut mengangsur pinjaman, maka jumlah pinjaman akan berkurang sesuai dengan nominal yang diangsur. Buatlah class Anggota tersebut, berikan atribut, method dan konstruktor sesuai dengan kebutuhan. Uji dengan TestKoperasi berikut ini untuk memeriksa apakah class Anggota yang anda buat telah sesuai dengan yang diharapkan.

public class TestKoperasi

{

public static void main(String[] args)

{

Anggota donny = new Anggota("111333444", "Donny", 5000000);

System.out.println("Nama Anggota: " + donny.getNama()); System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());

System.out.println("\nMeminjam uang 10.000.000..."); donny.pinjam(10000000);

System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());

System.out.println("\nMeminjam uang 4.000.000..."); donny.pinjam(4000000);

System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());

System.out.println("\nMembayar angsuran 1.000.000"); donny.angsur(1000000);

System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());

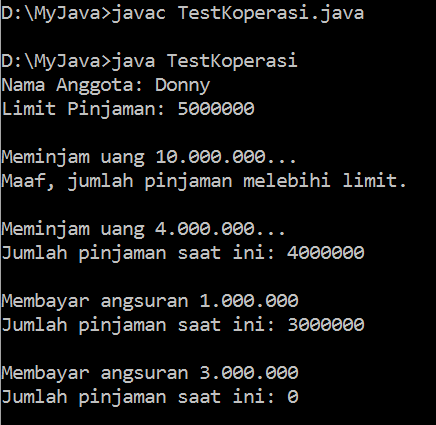
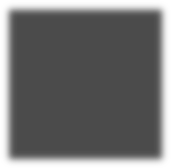
System.out.println("\nMembayar angsuran 3.000.000"); donny.angsur(3000000);

System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());

}

}

Hasil yang diharapkan:



package Tugas;

public class Anggota {

private String noKtp;

private String nama;

private int limit;

private int pinjaman;

Anggota(String noKtp, String nama, int limit){

this.noKtp = noKtp;

this.nama = nama;

this.limit = limit;

this.pinjaman = 0;

}

public String getNama(){

return nama;

}

public int getLimitPinjaman(){

return limit;

}

public void pinjam(int uang){

if (uang > limit){

System.out.println("Maaf, jumlah pinjaman melebihi limit.");

} else {

pinjaman += uang;

}

}

public int getJumlahPinjaman(){

return pinjaman;

}

public void angsur(int uang){

if (pinjaman < uang){

System.out.println("Pinjaman anda lebih kecil dari uang yang anda masukkan");

} else {

pinjaman -= uang;

}

}

}

* 1. Modifikasi soal no. 4 agar nominal yang dapat diangsur minimal adalah 10% dari jumlah pinjaman saat ini. Jika mengangsur kurang dari itu, maka muncul peringatan “Maaf, angsuran harus 10% dari jumlah pinjaman”.

else if(uang < (0.1 \* pinjaman)){

System.out.println("Maaf, angsuran harus 10% dari jumlah pinjaman");

}

* 1. Modifikasi class TestKoperasi, agar jumlah pinjaman dan angsuran dapat menerima input dari console.

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int uang, option;

System.out.println("Masukkan Identitas Nasabah\nNomor KTP: ");

String noKtp = sc.nextLine();

System.out.println("Nama : ");

String nama = sc.nextLine();

Anggota donny = new Anggota(noKtp, nama, 5000000);

System.out.println("Nama Anggota: " + donny.getNama());

System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());

do{

System.out.println("Menu\n1. Pinjam uang\n2. Bayar angsuran\n3. Keluar");

option = sc.nextInt();

switch(option){

case 1:

System.out.print("Nominal pinjam: ");

uang = sc.nextInt();

System.out.println("\nMeminjam uang " + uang + "...");

donny.pinjam(uang);

System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());

break;

case 2:

System.out.print("Nominal angsuran: ");

uang = sc.nextInt();

System.out.println("\nMembayar angsuran " + uang);

donny.angsur(uang);

System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());

break;

default:

option = 3;

}

} while(option < 3);

}