



NAMA : ANA QONITAH MUNAWWAROH
NIM : 2041720118
KELAS : 2C / 03
MATERI : LAPORAN PRAKTIKUM JOBSHEET 3 PRAKPBO

3.3 PERTANYAAN

1. Pada class TestMobil, saat kita menambah kecepatan untuk pertama kalinya, mengapa muncul peringatan “Kecepatan tidak bisa bertambah karena Mesin Off!”? [1] [SEP]

JAWAB : Kecepatan tidak bisa bertambah karena Mesin Off dikarenakan terjadi perubahan pada access modifier kecepatan dan kontakOn menjadi private. Dan pada class objek motor, nilai pada kontakOn masih bernilai false, sedangkan syarat untuk menambah kecepatan (dengan menggunakan if) adalah kontakOn harus bernilai true terlebih dahulu.

2. Mengapa atribut kecepatan dan kontakOn diset private? [1] [SEP]

JAWAB : Karena agar atribut kecepatan dan kontakOn tidak bisa diakses oleh pengguna dan diganti nilainya secara sembarangan. Dan juga agar atribut tersebut hanya dapat diakses didalam class Motor saja.

3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

JAWAB :

Kode program sebelum perubahan :

```
23 public void tambahkanKecepatan(){
24     if(kontakOn == true){
25         kecepatan += 5;
26     }
27     else{
28         System.out.println("Kecepatan tidak bisa bertambah karena mesin off!\n");
29     }
30 }
```

Kode program setelah perubahan :

```
23 public void tambahkanKecepatan(){
24     if(kecepatan == 100){
25         System.out.println("Kecepatan sudah maksimal, kurangi kecepatan Anda!");
26     }
27     else{
28         if(kontakOn == true){
29             kecepatan +=50;
30         } else {
31             System.out.println("Kecepatan tidak bisa bertambah karena Mesin Off!\n");
32         }
33     }
34 }
```



NAMA : ANA QONITAH MUNAWWAROH
NIM : 2041720118
KELAS : 2C / 03
MATERI : LAPORAN PRAKTIKUM JOBSHEET 3 PRAKPBO

Hasil running program :

```
run:
Mesin Off
Kecepatan: 0
=====
Kecepatan tidak bisa bertambah karena Mesin Off!

Mesin On
Kecepatan: 0
=====
Mesin On
Kecepatan: 50
=====
Mesin On
Kecepatan: 100
=====
Kecepatan sudah maksimal, kurangi kecepatan Anda!
Mesin On
Kecepatan: 100
=====
Mesin Off
Kecepatan: 0
=====
BUILD SUCCESSFUL (total time: 0 seconds)
```

3.6 PERTANYAAN – PERCOBAAN 3 DAN 4

1. Apa yang dimaksud getter dan setter?

JAWAB :

Getter Setter adalah sejenis method yang bertujuan untuk mengeset dan mendapatkan value dari sebuah atribut.

- Getter sendiri berguna untuk mendapatkan value dari atribut dan harus ada return value
- Setter sendiri berguna untuk mengeset value dari atribut dan tidak memiliki return value

2. Apa kegunaan dari method getSimpanan()?

JAWAB :

getSimpanan() bertujuan untuk mendapatkan value dari atribut simpanan yang terdapat di class Anggota.



NAMA : ANA QONITAH MUNAWWAROH
NIM : 2041720118
KELAS : 2C / 03
MATERI : LAPORAN PRAKTIKUM JOBSHEET 3 PRAKPBO

3. Method apa yang digunakan untk menambah saldo?

JAWAB :

Method yang digunakan untuk menambah saldo adalah method setor()

4. Apa yang dimaksud konstruktor?

JAWAB :

Konstruktor adalah sejenis method yang digunakan saat pembuatan objek (instansiasi)
Ciri-ciri konstruktor :

- tidak memiliki return value dan tidak memiliki tipe data method
- Nama method sama dengan nama class
- Hanya bisa dijalankan saat proses instansiasi
- Tidak dapat diset dengan access modifier

5. Sebutkan aturan dalam membuat konstruktor?

JAWAB : <http://bahasajava.com/9-aturan-contoh-penggunaan-konstruktor-pada-program-java/>

- a. Konstruktor dapat dioverload, yaitu satu class dapat memiliki lebih dari satu konstruktor asal parameternya berbeda.
- b. Default konstruktor diberikan oleh compiler java, sebenarnya compiler java membuat default constructor untuk class yang tidak memiliki parameter.
- c. Default konstruktor tidak diberikan oleh compiler java ketika sebuah class memiliki konstruktor.
- d. Hanya compiler yang membuat default konstruktor, jika Anda membuat konstruktor pada class yang sama persis dengan default konstruktor, maka itu tidak disebut default konstruktor.
- e. Suatu konstruktor tidak dapat diwariskan (inherited), method dapat diwariskan dari superclass ke subclassnya, namun konstruktor tidak dapat diwariskan.
- f. Konstruktor dapat bertipe private.
- g. Konstruktor akan memiliki access modifier yang sama dengan access modifier classnya.
- h. Konstruktor default dari superclass akan dipanggil oleh konstruktor subclassnya, compiler pada java akan secara otomatis memanggil konstruktor superclass ketika pada subclass dilakukan pemanggilan menggunakan keyword super().
- i. Super() atau this() adalah pernyataan pertama yang harus dipanggil dalam konstruktor.



NAMA : ANA QONITAH MUNAWWAROH
NIM : 2041720118
KELAS : 2C / 03
MATERI : LAPORAN PRAKTIKUM JOBSHEET 3 PRAKPBO

6. Apakah boleh konstruktor bertipe private?

JAWAB :

Konstruktor boleh bertipe private. Namun, konstruktor ini hanya akan dapat diakses melalui class yang sama saja dengan class tempat konstruktor berada, sehingga instansiasi objek hanya dapat dilakukan di class yang sama dan objek tidak dapat lebih dari satu buah. Namun, umumnya konstruktor bersifat publik, agar kita dapat mengaksesnya dari luar dan menginstansiasi objek lebih dari satu.

7. Kapan menggunakan parameter dengan passing parameter?

JAWAB :

Ketika menggunakan method yang digunakan untuk mengubah nilai atribut yang sudah memiliki nilai terlebih dahulu. Sedangkan passing parameter digunakan untuk melakukan instansiasi objek pada konstruktor.

8. Apa perbedaan atribut class dan instansiasi atribut?

JAWAB :

- Atribut class adalah atribut yang terdapat pada suatu class
- Instansiasi atribut adalah metode untuk membuat atribut pada suatu class

9. Apa perbedaan class method dan instansiasi method?

JAWAB :

- Class method adalah method yang terdapat pada suatu class
- Instansiasi method adalah metode untuk membuat sebuah method pada suatu class



NAMA : ANA QONITAH MUNAWWAROH
NIM : 2041720118
KELAS : 2C / 03
MATERI : LAPORAN PRAKTIKUM JOBSHEET 3 PRAKPBO

5. TUGAS

1. Cobalah program dibawah ini dan tuliskan hasil outputnya

JAWAB :

Kode program EncapTest :

```
12 public class EncapTest {  
13     public static void main(String[] args){  
14         EncapDemo encap = new EncapDemo();  
15         encap.setName("James");  
16         encap.setAge(35);  
17  
18         System.out.println("Name : "+encap.getName());  
19         System.out.println("Age : "+encap.getAge());  
20     }  
21 }
```

Kode program EncapDemo :

```
12 public class EncapDemo {  
13     private String name;  
14     private int age;  
15  
16     public String getName(){  
17         return name;  
18     }  
19  
20     public void setName(String newName){  
21         name = newName;  
22     }  
23  
24     public int getAge(){  
25         return age;  
26     }  
27  
28     public void setAge(int newAge){  
29         if(newAge > 30){  
30             age = 30;  
31         }  
32         else {  
33             age = newAge;  
34         }  
35     }  
36 }
```

```
run:  
Name : James  
Age : 30  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Hasil output program :



NAMA : ANA QONITAH MUNAWWAROH
NIM : 2041720118
KELAS : 2C / 03
MATERI : LAPORAN PRAKTIKUM JOBSHEET 3 PRAKPBO

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

JAWAB :

```
28 public void setAge(int newAge) {  
29     if(newAge > 30) {  
30         age = 30;  
31     }  
}
```

Karena pada method setAge(), dilakukan pemeriksaan yaitu parameter yang berisi nilai baru dari pemanggilan method setAge(), pada method main dilakukan pemanggilan method setAge() dengan parameter yang bernilai 35 dan dilakukan pemeriksaan pada fungsi if, jika variable newAge nilainya lebih dari 30 maka atribut age pada class EncapDemo akan diset 30.

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

JAWAB :

Kode program setelah diganti maksimal 30 dan minimal 18

```
28 public void setAge(int newAge) {  
29     if(newAge > 30) {  
30         age = 30;  
31     }  
32     else if (newAge < 18) {  
33         age = 18;  
34     }  
35     else {  
36         age = newAge;  
37     }  
38 }  
  
12 public class EncapTest {  
13     public static void main(String[] args) {  
14         EncapDemo encap = new EncapDemo();  
15         encap.setName("James");  
16         encap.setAge(10);  
17     }  
18 }
```

Hasil running program :

```
run:  
Name : James  
Age : 18  
BUILD SUCCESSFUL (total time: 0 seconds)
```



NAMA : ANA QONITAH MUNAWWAROH
NIM : 2041720118
KELAS : 2C / 03
MATERI : LAPORAN PRAKTIKUM JOBSHEET 3 PRAKPBO

4. Pada sebuah sistem informasi koperasi simpan pinjam, terdapat class Anggota yang memiliki atribut antara lain nomor KTP, nama, limit pinjaman, dan jumlah pinjaman. Anggota dapat meminjam uang dengan batas limit pinjaman 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.

JAWAB :

Class Anggota

```
1 package Pertemuan_3;
2
3 public class Anggota {
4     private String noKtp;
5     private String nama;
6     private int limitPinjaman;
7     private int jumlahPinjaman;
8
9     Anggota(String noKtp, String nama, int limitPinjaman){
10         this.noKtp = noKtp;
11         this.nama = nama;
12         this.limitPinjaman = limitPinjaman;
13         jumlahPinjaman = 0;
14     }
15     public void setNoKtp(String noKtp){
16         this.noKtp = noKtp;
17     }
18     public void setName(String nama){
19         this.nama = nama;
20     }
21
22     public String getNoKtp(String noKtp){
23         return noKtp;
24     }
25     public String getName(){
26         return nama;
27     }
28     public int getLimitPinjaman(){
29         return limitPinjaman;
30     }
31     public int getJumlahPinjaman(){
32         return jumlahPinjaman;
33     }
34
35     public void angsur(int angsur){
36         jumlahPinjaman -= angsur;
37     }
38     public void pinjam(int jumlahPeminjaman){
39         if(limitPinjaman > jumlahPeminjaman){
40             this.jumlahPinjaman += jumlahPeminjaman;
41             limitPinjaman -= jumlahPeminjaman;
42         } else {
43             System.out.println("Maaf, jumlah pinjaman melebihi limit!\n");
44         }
45     }
46 }
```



NAMA : ANA QONITAH MUNAWWAROH
NIM : 2041720118
KELAS : 2C / 03
MATERI : LAPORAN PRAKTIKUM JOBSHEET 3 PRAKPBO

Class Test Koperasi

```
6 package Pertemuan_3;
7
8 /**
9  *
10  * @author Lenovo
11  */
12 public class TestKoperasi {
13     public static void main(String[] args){
14         Anggota donny = new Anggota("111333444", "Donny", 5000000);
15
16         System.out.println("Nama Anggota: " + donny.getNama());
17         System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());
18         System.out.println("\nMeminjam uang 10.000.000...");
19         donny.pinjam(10000000);
20
21         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
22         System.out.println("\nMeminjam uang 4.000.000...");
23         donny.pinjam(4000000);
24
25         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
26         System.out.println("\nMembayar angsuran 1.000.000");
27         donny.angsur(1000000);
28
29         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
30         System.out.println("\nMembayar angsuran 3.000.000");
31         donny.angsur(3000000);
32         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
33     }
34 }
```

Hasil running program :

```
run:
Nama Anggota: Donny
Limit Pinjaman: 5000000

Meminjam uang 10.000.000...
Maaf, jumlah pinjaman melebihi limit!

Jumlah pinjaman saat ini: 0

Meminjam uang 4.000.000...
Jumlah pinjaman saat ini: 4000000

Membayar angsuran 1.000.000
Jumlah pinjaman saat ini: 3000000

Membayar angsuran 3.000.000
Jumlah pinjaman saat ini: 0
BUILD SUCCESSFUL (total time: 0 seconds)
```




NAMA : ANA QONITAH MUNAWWAROH
NIM : 2041720118
KELAS : 2C / 03
MATERI : LAPORAN PRAKTIKUM JOBSHEET 3 PRAKPBO

5. 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".

JAWAB :

Perubahan pada class testKoperasi

```
6 package Pertemuan_3;
7
8 /**
9  *
10  * @author Lenovo
11  */
12 public class TestKoperasi {
13     public static void main(String[] args){
14         Anggota donny = new Anggota("111333444", "Donny", 5000000);
15
16         System.out.println("Nama Anggota: " + donny.getNama());
17         System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());
18         System.out.println("\nMeminjam uang 10.000.000...");
19         donny.pinjam(10000000);
20
21         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
22         System.out.println("\nMeminjam uang 4.000.000...");
23         donny.pinjam(4000000);
24
25         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
26         System.out.println("\nMembayar angsuran 1.000.000");
27         donny.angsur(1000000);
28
29         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
30         System.out.println("\nMembayar angsuran 3.000.000");
31         donny.angsur(2000000);
32         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
33     }
34 }
```

Perubahan pada method angsur() :

```
35 public void angsur(int angsur){
36     double minimum = jumlahPinjaman * 0.1;
37     if(angsur > (int)minimum){
38         jumlahPinjaman -= angsur;
39     } else {
40         System.out.println("Maaf, anggaran harus 10% dari jumlah pinjaman!");
41     }
42 }
43 }
```



NAMA : ANA QONITAH MUNAWWAROH
NIM : 2041720118
KELAS : 2C / 03
MATERI : LAPORAN PRAKTIKUM JOBSHEET 3 PRAKPBO

Hasil running program :

```
run:
Nama Anggota: Donny
Limit Pinjaman: 5000000

Meminjam uang 10.000.000...
Maaf, jumlah pinjaman melebihi limit!

Jumlah pinjaman saat ini: 0

Meminjam uang 4.000.000...
Jumlah pinjaman saat ini: 4000000

Membayar angsuran 1.000.000
Jumlah pinjaman saat ini: 3000000

Membayar angsuran 3.000.000
Maaf, anggaran harus 10% dari jumlah pinjaman!
Jumlah pinjaman saat ini: 3000000
BUILD SUCCESSFUL (total time: 0 seconds)
```

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

JAWAB : Perubahan pada class testKoperasi

```
6 package Pertemuan_3;
7 import java.util.Scanner;
8 /**
9  *
10 * @author Lenovo
11 */
12 public class TestKoperasi {
13     public static void main(String[] args){
14         Anggota donny = new Anggota("111333444", "Donny", 5000000);
15         Scanner ana = new Scanner(System.in);
16
17         System.out.println("Nama Anggota: " + donny.getNama());
18         System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());
19
20         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
21         System.out.print("Masukkan jumlah dari peminjaman: ");
22         int pinjam = ana.nextInt();
23         donny.pinjam(pinjam);
24
25         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
26         System.out.print("Masukkan jumlah dari angsuran: ");
27         int angsur = ana.nextInt();
28         donny.angsur(angsur);
29         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
30     }
31 }
```

```
run:
Nama Anggota: Donny
Limit Pinjaman: 5000000
Jumlah pinjaman saat ini: 0
Masukkan jumlah dari peminjaman: 4000000
Jumlah pinjaman saat ini: 4000000
Masukkan jumlah dari angsuran: 3500000
Jumlah pinjaman saat ini: 500000
BUILD SUCCESSFUL (total time: 10 seconds)
```

Hasil running program :