

Nama: Rara Deninda Hurianto

NIM: 2041720110

Praktikum PBO

3.3 Pertanyaan

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

Jawab:

Ini terjadi pada class Motor karena nilai pada KontakOn masih bernilai false, sedangkan syarat untuk menambah kecepatan dengan metode if adalah kontakOn harus bernilai true terlebih dahulu

2. Mengapa atribut kecepatan dan kontakOn diset private?

Jawab:

Karena agar atribut kecepatan tidak bisa diakses oleh pengguna dan diganti nilainya secara sembarangan. Sehingga Ketika kita mencoba menambahkan kecepatan saat posisi kontak off maka muncul notifikasi bahwa mesin off, jika ingin menambah kecepatan maka harus memanggil method tambahKecepatan() beberapa kali dahulu.

3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

Jawab:

```
public void tambahKecepatan() {  
    if (kecepatan == 100){  
        System.out.println("Kecepatan mencapai batas maksimal, kurangi kecepatan anda!");  
    } else {  
        if (kontakOn == true) {  
            kecepatan += 5;  
        } else {  
            System.out.println("Kecepatan tidak bisa bertambah karena Mesin Off:\n");  
        }  
    }  
}
```

3.6 Pertanyaan – Percobaan 3 dan 4

1. Apa yang dimaksud getter dan setter?

Jawab:

Getter dan Setter adalah method untuk mengubah nilai dari atribut, sehingga nilai dari atribut ini tidak dapat diubah secara langsung (atribut private),

melainkan harus menggunakan Getter dan Setter yang nama methodnya telah ditentukan dan dapat dipanggil dari class lain.

2. Apa kegunaan dari method `getSimpanan()`?

Jawab:

Kegunaan dari method `getSimpanan()` adalah mengetahui nilai dari atribut simpanan yang terdapat pada class Anggota

3. Method apa yang digunakan untuk menambah saldo?

Jawab:

Untuk menambah saldo dapat digunakan method `setor()`

4. Apa yang dimaksud konstruktor?

Jawab:

Konstruktor adalah method untuk melakukan instansiasi objek yang baru ini memiliki nilai atribut yang spesifik pada parameternya. Apabila nilai atribut dari objek yang baru adalah bernilai sama dengan objek lain, maka tidak perlu dimasukkan pada parameter konstruktor

5. Sebutkan aturan dalam membuat konstruktor?

Jawab:

- Nama konstruktor harus sama dengan nama class nya
- Konstruktor tidak memiliki return
- Tidak boleh menggunakan modifier `abstract`, `static`, `final`, dan `synchronized`

6. Apakah boleh konstruktor bertipe `private`?

Jawab:

Boleh, konstruktor dapat 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 `public`, 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 class yang menampung atribut suatu objek yang akan digunakan pada main class, sedangkan instansiasi atribut adalah atribut yang didefinisikan dalam suatu method.

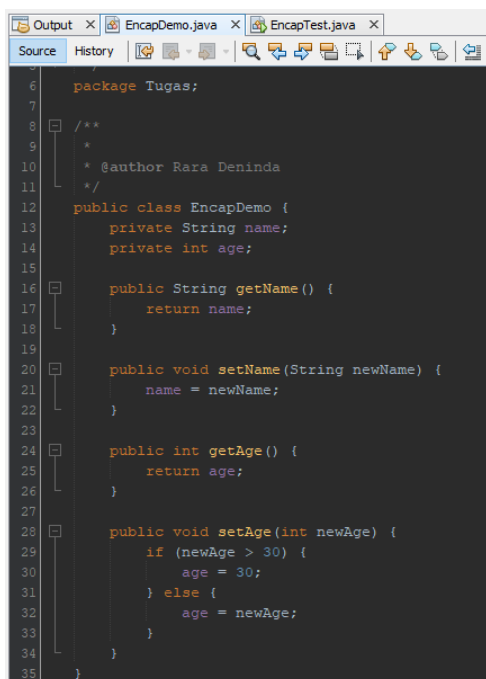
9. Apa perbedaan class method dan instansiasi method?

Jawab:

Class method adalah class yang menampung method method yang akan digunakan, sedangkan instansiasi method adalah metode untuk membuat sebuah method pada suatu class.

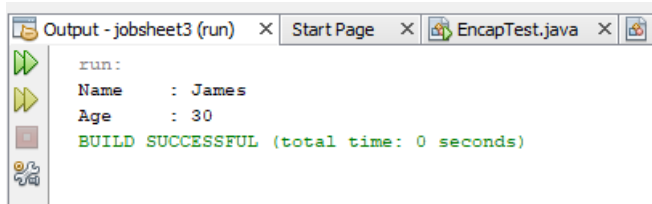
TUGAS

1. Kode dan output



```
1 package Tugas;
2
3 /**
4  *
5  * @author Rara Deninda
6  */
7 public class EncapDemo {
8     private String name;
9     private int age;
10
11     public String getName() {
12         return name;
13     }
14
15     public void setName(String newName) {
16         name = newName;
17     }
18
19     public int getAge() {
20         return age;
21     }
22
23     public void setAge(int newAge) {
24         if (newAge > 30) {
25             age = 30;
26         } else {
27             age = newAge;
28         }
29     }
30 }
31
32
33
34
35
```

```
5  L  */
6  package Tugas;
7
8  /**
9   *
10  * @author Rara Deninda
11  */
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\t: " + encap.getName());
19          System.out.println("Age\t: " + encap.getAge());
20      }
21  }
```



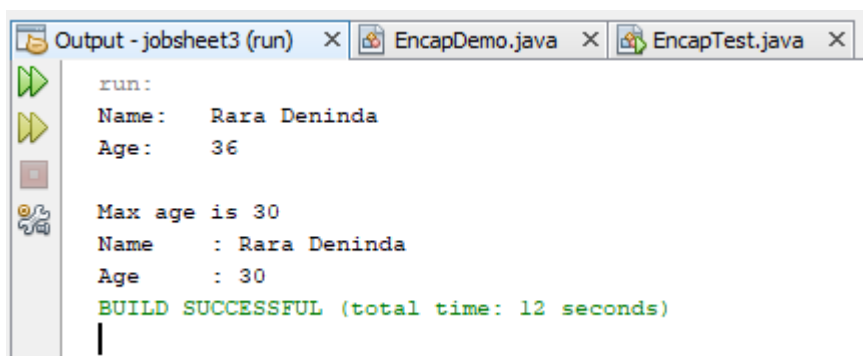
2. Pada program diatas, pada class EncapTest kita mengeset age dengan nilai 35, namun pada saat ditampilkan ke layar nilainya 30, jelaskan mengapa.

Jawab:

Pada method setAge() berparameter int newAge yang dapat diakses di class main dengan mengisi argument bertipe int. didalam method in terdapat kode pemilihan (if else) apabila nilai dari newAge lebih dari 30 maka nilai dari age = 30, apabila pemilihan awal bernilai false maka nilai dari age = nilai newAge. Itulah mengapa outputnya = 30.

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

Jawab:



```
Output - jobsheet3 (run) x EncapDemo.java x EncapTest.java x
run:
Name:   Rara Deninda
Age:    17

Min age is 18
Name    : Rara Deninda
Age     : 18
BUILD SUCCESSFUL (total time: 8 seconds)
```

```
Output - jobsheet3 (run) x EncapDemo.java x EncapTest.java x
run:
Name:   Rara Deninda
Age:    24

Name    : Rara Deninda
Age     : 24
BUILD SUCCESSFUL (total time: 7 seconds)
|
```

4. Uji dengan TestKoperasi berikut ini untuk memeriksa apakah class Anggota yang anda buat telah sesuai dengan yang diharapkan.

Jawab:

```
12 public class Anggota {
13     private String ktpNo, nama;
14     private int limit, jumlah;
15
16     Anggota(String name, String ktpNo, int limit) {
17         this.nama = name;
18         this.ktpNo = ktpNo;
19         this.limit = limit;
20     }
21     public String getName() {
22         return nama;
23     }
24     public int getLimitPinjaman() {
25         return limit;
26     }
27     public void pinjam(int pinjam) {
28         if (pinjam < limit) {
29             jumlah = pinjam;
30         } else {
31             System.out.print("Maaf, pinjaman melebihi limit!\n");
32         }
33     }
34     public void angsur(int angsur) {
35         jumlah = jumlah - angsur;
36     }
37     public int getJumlahPinjaman() {
38         return jumlah;
39     }
40 }
41
```

```

11 public class TestKoperasi {
12     public static void main(String[] args) {
13         Anggota donny = new Anggota("Donny", "111333444", 5000000);
14
15         System.out.println("Nama Anggota\t: " + donny.getName());
16         System.out.println("Limit Pinjaman\t: " + donny.getLimitPinjaman());
17
18         System.out.println("\nMeminjam uang 10.000.000...");
19         donny.pinjam(10000000);
20         System.out.println("Jumlah pinjaman saat ini : " + donny.getJumlahPinjaman());
21
22         System.out.println("\nMeminjam uang 4.000.000...");
23         donny.pinjam(4000000);
24         System.out.println("Jumlah pinjaman saat ini : " + donny.getJumlahPinjaman());
25
26         System.out.println("\nMembayar angsuran 1.000.000...");
27         donny.angsur(1000000);
28         System.out.println("Jumlah pinjaman saat ini : " + donny.getJumlahPinjaman());
29
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 }

```

```

Output - jobsheet3 (run)  EncapDemo.java  EncapTestJa
run:
Nama Anggota      : Donny
Limit Pinjaman    : 5000000

Meminjam uang 10.000.000...
Maaf, 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)

```

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:

```

34 public void angsur(int angsur){
35     int angsuran;
36     angsuran = (int) (jumlah * 0.1);
37     if (angsur >= angsuran){
38         jumlah = jumlah - angsur;
39     } else {
40         System.out.println("Maaf, Angsuran Harus 10% dari Jumlah Pinjaman");
41     }
42 }

```

```

Output - jobsheet3 (run) X EncapDemo.java X EncapTest.java X Angg
run:
Nama Anggota : Donny
Limit Pinjaman : 5000000

Meminjam uang 10.000.000...
Maaf, 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 100000
Maaf, Angsuran Harus 10% dari Jumlah Pinjaman
Jumlah pinjaman saat ini : 3000000
BUILD SUCCESSFUL (total time: 3 seconds)

```

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

Jawab:

```

7 import java.util.Scanner;
8
9
10 * @author Rara Deninda
11 */
12 public class TestKoperasi {
13     public static void main(String[] args) {
14         Scanner rara = new Scanner(System.in);
15         Anggota donny = new Anggota("Donny", "111333444", 5000000);
16
17         System.out.println("Nama Anggota\t: " + donny.getName());
18         System.out.println("Limit Pinjaman\t: " + donny.getLimitPinjaman());
19
20         int menu, pinjaman, angsuran;
21
22         do{
23             System.out.println("=====");
24             System.out.println("1. Peminjaman");
25             System.out.println("2. Angsuran");
26             System.out.println("3. Selesai\n");
27             System.out.print("Menu : ");
28             menu = rara.nextInt();
29             System.out.println("=====");
30             if(menu == 1){
31                 System.out.print("Nominal Pinjaman : ");
32                 pinjaman = rara.nextInt();
33                 donny.pinjam(pinjaman);
34                 System.out.println("Meminjam Uang "+pinjaman);
35                 System.out.println("Jumlah pinjaman : " + donny.getJumlahPinjaman());
36             } else if(menu == 2){
37                 System.out.print("Nominal Angsuran : ");
38                 angsuran = rara.nextInt();
39                 donny.angsur(angsuran);
40                 System.out.println("Membayar Angsuran "+angsuran);
41                 System.out.println("Jumlah pinjaman : " + donny.getJumlahPinjaman());
42             }
43         } while(donny.getJumlahPinjaman() != 0 && menu != 3);
44     }

```

```
Output - jobsheet3 (run) X EncapDemo.java X EncapTest.java X
run:
Nama Anggota : Donny
Limit Pinjaman : 5000000
=====
1. Peminjaman
2. Angsuran
3. Selesai

Menu : 1
=====
Nominal Pinjaman : 250000
Meminjam Uang 250000
Jumlah pinjaman : 250000
=====
1. Peminjaman
2. Angsuran
3. Selesai

Menu : 2
=====
Nominal Angsuran : 100000
Membayar Angsuran 100000
Jumlah pinjaman : 150000
=====
1. Peminjaman
2. Angsuran
3. Selesai

Menu : 3
=====
BUILD SUCCESSFUL (total time: 17 seconds)
```