

PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK

“Enkapsulasi”



DOSEN PEMBIMBING

Septian Enggar Sukmana, S.Pd., M.T.

Disusun oleh :

Muhammad Reza Khatami

(2041720076)

PROGRAM STUDI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG 2021

Jobsheet 3

Praktikum!

1. Percobaan 1

- Motor.java

```
1 package Praktikum;
2
3 public class Motor {
4
5     private int kecepatan = 0;
6     private boolean kontakOn = false;
7
8     public void nyalakanMesin() {
9         kontakOn = true;
10    }
11
12    public void matikanMesin() {
13        kontakOn = false;
14        kecepatan = 0;
15    }
16
17    public void tambahKecepatan() {
18        if (kontakOn == true) {
19            kecepatan += 5;
20        } else {
21            System.out.println("Kecepatan tidak bisa bertambah karena Mesin Off! \n");
22        }
23    }
24
25    public void kurangiKecepatan() {
26        if (kontakOn == true) {
27            kecepatan -= 5;
28        } else {
29            System.out.println("Kecepatan tidak bisa berkurang karena mesin Off! \n");
30        }
31    }
32
33    public void printStatus() {
34        if (kontakOn == true) {
35            System.out.println("Kontak On");
36        } else {
37            System.out.println("Kontak Off");
38        }
39        System.out.println("Kecepatan " + kecepatan + "\n");
40    }
41 }
```

- MotorDemo.java

```
1 package Praktikum;
2
3 public class MotorDemo {
4     public static void main(String[] args) {
5         Motor motor = new Motor();
6         motor.printStatus();
7         motor.tambahKecepatan();
8
9         motor.nyalakanMesin();
10        motor.printStatus();
11
12        motor.tambahKecepatan();
13        motor.printStatus();
14
15        motor.tambahKecepatan();
16        motor.printStatus();
17
18        motor.tambahKecepatan();
19        motor.printStatus();
20
21        motor.matikanMesin();
22        motor.printStatus();
23    }
24 }
```

- Run Program

```

run:
Kontak Off
Kecepatan 0

Kecepatan tidak bisa bertambah karena Mesin Off!

Kontak On
Kecepatan 0

Kontak On
Kecepatan 5

Kontak On
Kecepatan 10

Kontak On
Kecepatan 15

Kontak Off
Kecepatan 0

BUILD SUCCESSFUL (total time: 0 seconds)

```

2. Percobaan 2

Pertanyaan!

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

Jawab :

Dikarenakan nilai pada kontakOn awal dideklarasikan pada class Motor diawal masih bernilai false, sedangkan syarat menambah kecepatan motor yaitu mesin harus dinyalakan terlebih dahulu atau Mesin on, seperti yang sudah di deklarasikan pada kode program (sesuai dengan metode if yang ada di method tambahKecepatan() bahwa kontakOn == true)

2. Mengapat atribut kecepatan dan kontakOn diset private?

Jawab :

atribut kecepatan dan kontakOn diset private karena awalnya akses ke atribut motor tidak terkontrol, maka dari itu di set private supaya tidak bisa diakses oleh class lain. Sehingga Ketika kita mencoba menambahkan kecepatam saat posisi kontak off maka muncul notifikasi bahwa mesin off, jika ingin menambah kecepatan maka harus memanggil method tambahKecepatan() beberapa kali dahulu.

3. Ubahlah class Motor sehingga kecepatan maksimal adalah 100!

Jawab :

```

17 public void tambahKecepatan() {
18     if (kecepatan == 100){
19         System.out.println("Kecepatan Mencapai batas maksimal, Kurangi kecepatan anda!! \n");
20     } else if (kontakOn == true) {
21         kecepatan += 5;
22     } else {
23         System.out.println("Kecepatan tidak bisa bertambah karena Mesin Off! \n");
24     }
25 }

```

3. Percobaan 3 dan 4

Pertanyaan!

1. Apa yang dimaksud getter dan setter?

Jawab :

Method setter dan

getter adalah dua method yang tugasnya untuk mengambil dan mengisi data ke dalam objek. 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. method setter adalah suatu method pada java yang digunakan untuk memberikan nilai pada variabel. Sedangkan method getter adalah method yang mengembalikan suatu nilai.

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 :

Method yang digunakan untuk menambahkan saldo adalah method setor()

4. Apa yang dimaksud konstruktor?

Jawab :

Konstruktor adalah merupakan suatu method yang akan memberikan nilai awal pada saat suatu objek dibuat. Pada saat program dijalankan, constructor akan langsung memberikan nilai awal pada saat perintah new, membuat suatu objek. Nama Constructor sama dengan nama Class, selain itu 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 :

- 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.
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 :

Penggunaan parameter dengan passing parameter adalah 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 :

Perbedaan atribut class dan instansiasi atribut adalah , atribut class adalah atribut pada suatu class, sedangkan instansiasi atribut adalah metode untuk membuat atribut pada suatu class.

9. Apa perbedaan class method dan instansiasi method?

Jawab :

Perbedaan class method dan instansiasi method adalah 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. Tugas 1

- **EncapDemo.java**

```

1  package TugasJS3;
2
3  public class EncapDemo {
4      private String name;
5      private int age;
6
7      public String getName(){
8          return name;
9      }
10
11     public void setName(String newName){
12         name = newName;
13     }
14
15     public int getAge(){
16         return age;
17     }
18
19     public void setAge(int newAge){
20         if(newAge > 30){
21             age = 30;
22         } else {
23             age = newAge;
24         }
25     }
26 }

```

- EncapTest.java

```

1  package TugasJS3;
2
3  public class EncapTest {
4      public static void main(String[] args) {
5          EncapDemo encap = new EncapDemo();
6          encap.setName("james");
7          encap.setAge(35);
8
9          System.out.println("Name : " + encap.getName());
10         System.out.println("Age : " + encap.getAge());
11     }
12 }

```

- Run Program

```

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

```

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

Jawab :

Karena di program terdapat statement bahwa jika usia lebih dari 30, maka umur = 30, oleh karena itu jika kita set umur 35 maka munculnya akan 30 juga.

- Tugas 3

- EncapDemo.java

```

1  package TugasJS3;
2
3  public class EncapDemo {
4      private String name;
5      private int age;
6
7      public String getName(){
8          return name;
9      }
10
11     public void setName(String newName){
12         name = newName;
13     }
14
15     public int getAge(){
16         return age;
17     }
18
19     public void setAge(int newAge){
20         if(newAge > 30){
21             System.out.println("Umur anda tidak sesuai");
22         } else if (newAge < 18){
23             System.out.println("Anda belum cukup umur");
24         } else {
25             age = newAge;
26         }
27     }
28 }

```

- EncapTest.java

```

1  package TugasJS3;
2
3  public class EncapTest {
4      public static void main(String[] args) {
5          EncapDemo encap = new EncapDemo();
6          encap.setName("james");
7          encap.setAge(15);
8
9          System.out.println("Name      : " + encap.getName());
10         System.out.println("Age       : " + encap.getAge());
11     }
12 }

```

- Run Program

```

run:
Anda belum cukup umur
Name      : james
Age       : 0
BUILD SUCCESSFUL (total time: 0 seconds)

```

4. Tugas 4

- Anggota.java

```
1 package TugasJS3;
2
3 public class Anggota {
4
5     private String nomorKTP, nama;
6     private int limitPinjaman, jumlahPinjaman;
7
8     public Anggota(String nomorKTP, String nama, int limitPinjaman) {
9         this.nomorKTP = nomorKTP;
10        this.nama = nama;
11        this.limitPinjaman = limitPinjaman;
12    }
13
14    public String getNomorKTP() {
15        return nomorKTP;
16    }
17
18    public void setNomorKTP(String nomorKTP) {
19        this.nomorKTP = nomorKTP;
20    }
21
22    public String getName() {
23        return nama;
24    }
25
26    public void setName(String nama) {
27        this.nama = nama;
28    }
29
30    public double getLimitPinjaman() {
31        return limitPinjaman;
32    }
33
34    public void setLimitPeminjaman(int limitPeminjaman) {
35        this.limitPinjaman = limitPeminjaman;
36    }
37
38    public double getJumlahPinjaman() {
39        return jumlahPinjaman;
40    }
41
42    public void setJumlahPinjaman(int jumlahPinjaman) {
43        this.jumlahPinjaman = jumlahPinjaman;
44    }
45
46    public int pinjam(int jumlahPinjam) {
47        jumlahPinjaman += jumlahPinjam;
48        if (jumlahPinjaman > getLimitPinjaman()) {
49            System.out.println("Maaf, jumlah pinjaman melebihi limit!");
50            jumlahPinjaman -= jumlahPinjam;
51        }
52        return jumlahPinjaman;
53    }
54
55    public int angsur(int jumlahAngsur) {
56        jumlahPinjaman -= jumlahAngsur;
57        return jumlahPinjaman;
58    }
59 }
```

- TestKoperasi.java


```

1 package TugasJS3;
2
3 public class TestKoperasi {
4
5     public static void main(String[] args) {
6         Anggota donny = new Anggota("111333444", "Donny", 5000000);
7         System.out.println("Nama Anggota: " + donny.getNama());
8         System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());
9         System.out.println("\nMeminjam uang 10.000.000...");
10        donny.pinjam(10000000);
11        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
12        System.out.println("\nMeminjam uang 4.000.000...");
13        donny.pinjam(4000000);
14        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
15        System.out.println("\nMembayar angsuran 1.000.000");
16        donny.angsur(1000000);
17        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
18        System.out.println("\nMembayar angsuran 3.000.000");
19        donny.angsur(3000000);
20        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
21    }
22 }

```

- Run Program

```

run:
Nama Anggota: Donny
Limit Pinjaman: 5000000.0

Meminjam uang 10.000.000...
Maaf, jumlah pinjaman melebihi limit!
Jumlah pinjaman saat ini: 0.0

Meminjam uang 4.000.000...
Jumlah pinjaman saat ini: 4000000.0

Membayar angsuran 1.000.000
Jumlah pinjaman saat ini: 3000000.0

Membayar angsuran 3.000.000
Jumlah pinjaman saat ini: 0.0
BUILD SUCCESSFUL (total time: 0 seconds)

```

5. Tugas 5

- Anggota.java

```
1  package TugasJS3;
2
3  public class Anggota {
4
5      private String nomorKTP, nama;
6      private int limitPinjaman, jumlahPinjaman;
7
8      public Anggota(String nomorKTP, String nama, int limitPinjaman) {
9          this.nomorKTP = nomorKTP;
10         this.nama = nama;
11         this.limitPinjaman = limitPinjaman;
12     }
13
14     public String getNomorKTP() {
15         return nomorKTP;
16     }
17
18     public void setNomorKTP(String nomorKTP) {
19         this.nomorKTP = nomorKTP;
20     }
21
22     public String getName() {
23         return nama;
24     }
25
26     public void setName(String nama) {
27         this.nama = nama;
28     }
29
30     public double getLimitPinjaman() {
31         return limitPinjaman;
32     }
33
34     public void setLimitPeminjaman(int limitPeminjaman) {
35         this.limitPinjaman = limitPeminjaman;
36     }
37
38     public double getJumlahPinjaman() {
39         return jumlahPinjaman;
40     }
41
42     public void setJumlahPinjaman(int jumlahPinjaman) {
43         this.jumlahPinjaman = jumlahPinjaman;
44     }
45
46     public int pinjam(int jumlahPinjam) {
47         jumlahPinjaman += jumlahPinjam;
48         if (jumlahPinjaman > getLimitPinjaman()) {
49             System.out.println("Maaf, jumlah pinjaman melebihi limit!");
50             jumlahPinjaman -= jumlahPinjam;
51         }
52         return jumlahPinjaman;
53     }
54
55     public int angsur(int jumlahAngsur) {
56         if (jumlahAngsur < (0.1 * jumlahPinjaman)) {
57             System.out.println("Maaf, angsuran harus 10% dari jumlah pinjaman");
58             jumlahPinjaman += jumlahAngsur;
59         } else {
60             jumlahPinjaman -= jumlahAngsur;
61         }
62         return jumlahPinjaman;
63     }
64 }
65 }
```

- TestKoperasi.java

```

1 package TugasJS3;
2
3 public class TestKoperasi {
4
5     public static void main(String[] args) {
6         Anggota donny = new Anggota("111333444", "Donny", 5000000);
7         System.out.println("Nama Anggota: " + donny.getNama());
8         System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());
9         System.out.println("\nMeminjam uang 10.000.000...");
10        donny.pinjam(10000000);
11        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
12        System.out.println("\nMeminjam uang 4.000.000...");
13        donny.pinjam(4000000);
14        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
15        System.out.println("\nMembayar angsuran 1.000.000");
16        donny.angsur(500);
17        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
18        System.out.println("\nMembayar angsuran 3.000.000");
19        donny.angsur(300);
20        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
21    }
22 }

```

- Run Program

run:

Nama Anggota: Donny
Limit Pinjaman: 5000000.0

Meminjam uang 10.000.000...
Maaf, jumlah pinjaman melebihi limit!
Jumlah pinjaman saat ini: 0.0

Meminjam uang 4.000.000...
Jumlah pinjaman saat ini: 4000000.0

Membayar angsuran 1.000.000
Maaf, angsuran harus 10% dari jumlah pinjaman
Jumlah pinjaman saat ini: 4000500.0

Membayar angsuran 3.000.000
Maaf, angsuran harus 10% dari jumlah pinjaman
Jumlah pinjaman saat ini: 4000800.0

BUILD SUCCESSFUL (total time: 0 seconds)

6. Tugas 6

- TestKoperasi.java

```
1 package TugasJS3;
2
3 import java.util.Scanner;
4
5 public class TestKoperasi {
6
7     public static void main(String[] args) {
8         Scanner reza = new Scanner(System.in);
9         Anggota donny = new Anggota("111333444", "Donny", 5000000);
10        System.out.println("Nama Anggota: " + donny.getNama());
11        System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());
12        System.out.println("\nMeminjam uang 10.000.000...");
13        System.out.print("Masukkan Jumlah Pinjaman : ");
14        donny.pinjam(reza.nextInt());
15        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
16        System.out.println("\nMeminjam uang 4.000.000...");
17        System.out.print("Masukkan Jumlah Pinjaman : ");
18        donny.pinjam(reza.nextInt());
19        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
20        System.out.println("\nMembayar angsuran 1.000.000");
21        System.out.print("Masukkan Jumlah Angsuran : ");
22        donny.angsur(reza.nextInt());
23        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
24        System.out.println("\nMembayar angsuran 3.000.000");
25        System.out.print("Masukkan Jumlah Angsuran : ");
26        donny.angsur(reza.nextInt());
27        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
28    }
29 }
```

- Run Program

```
run:
Nama Anggota: Donny
Limit Pinjaman: 5000000.0

Meminjam uang 10.000.000...
Masukkan Jumlah Pinjaman : 10000000
Maaf, jumlah pinjaman melebihi limit!
Jumlah pinjaman saat ini: 0.0

Meminjam uang 4.000.000...
Masukkan Jumlah Pinjaman : 4000000
Jumlah pinjaman saat ini: 4000000.0

Membayar angsuran 1.000.000
Masukkan Jumlah Angsuran : 1000000
Jumlah pinjaman saat ini: 3000000.0

Membayar angsuran 3.000.000
Masukkan Jumlah Angsuran : 3000000
Jumlah pinjaman saat ini: 0.0
BUILD SUCCESSFUL (total time: 21 seconds)
```