

LAPORAN PRAKTIKUM
PERMROGRAMAN BERBASIS OBJEK
JOBSHEET 4
(Relasi Kelas)

Debora Febriana P. A
TI2B / 08
1941720156

Percobaan 1

1. Class Processor

```
1  package Jobsheet4.Percobaan1;
2  public class Processor {
3      private String merk;
4      private double cache;
5
6  Processor() {
7  }
8
9  Processor (String merk, double cache){
10     this.merk=merk;
11     this.cache=cache;
12 }
13
14 public void setmerk(String merk){
15     this.merk = merk;
16 }
17
18 public String getmerk(){
19     return merk;
20 }
21
22 public void setcache(double cache){
23     this.cache = cache;
24 }
25
26 public double getcache() {
27     return cache;
28 }
29
30 public void info() {
31     System.out.printf("Merk Processor = %s\n", merk);
32     System.out.printf("Cache Memory = %.2f\n", cache);
33 }
34 }
```

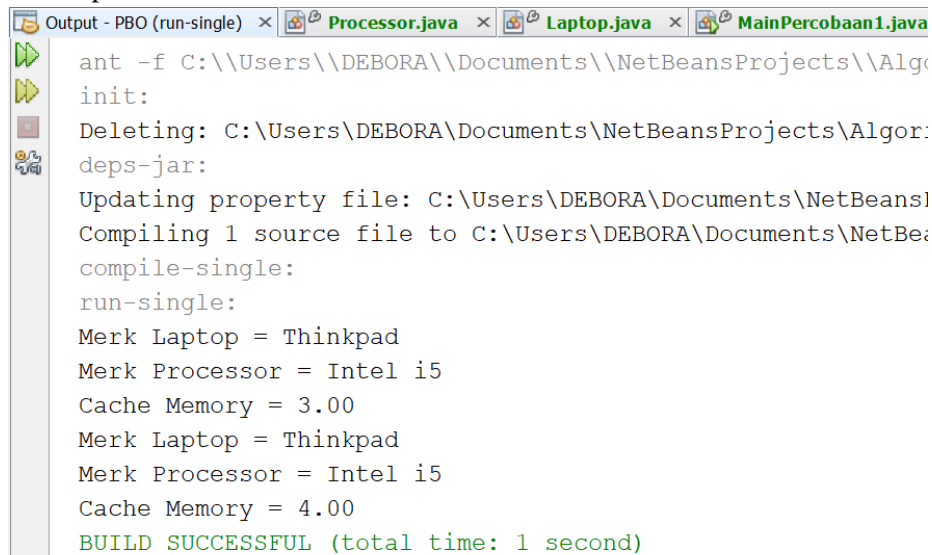
2. Class Laptop

```
1  package Jobsheet4.Percobaan1;
2  public class Laptop {
3      private String merk;
4      private Processor proc;
5
6      Laptop() {
7      }
8
9      public Laptop(String merk, Processor proc){
10         this.merk = merk;
11         this.proc = proc;
12     }
13
14     public void setmerk(String merk) {
15         this.merk = merk;
16     }
17
18     public String getmerk(){
19         return merk;
20     }
21
22     public void setproc(Processor proc){
23         this.proc = proc;
24     }
25
26     public Processor getproc(){
27         return proc;
28     }
29
30     public void info(){
31         System.out.println("Merk Laptop = "+merk);
32         proc.info();
33     }
34 }
```

3. Class MainPercobaan1

```
1  /*
2   * To change this license header, choose License Headers i
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package Jobsheet4.Percobaan1;
7  public class MainPercobaan1 {
8      public static void main(String args[]){
9          Processor p = new Processor("Intel i5",3);
10         Laptop L = new Laptop("Thinkpad",p);
11         L.info();
12
13         Processor p1 = new Processor();
14         p1.setmerk("Intel i5");
15         p1.setcache(4);
16         Laptop L1 = new Laptop();
17         L1.setmerk("Thinkpad");
18         L1.setproc(p1);
19         L1.info();
20     }
21 }
```

4. Ouput



```
Output - PBO (run-single) x Processor.java x Laptop.java x MainPercobaan1.java
ant -f C:\\Users\\DEBORA\\Documents\\NetBeansProjects\\Algo
init:
Deleting: C:\\Users\\DEBORA\\Documents\\NetBeansProjects\\Algor:
deps-jar:
Updating property file: C:\\Users\\DEBORA\\Documents\\NetBeansI
Compiling 1 source file to C:\\Users\\DEBORA\\Documents\\NetBea
compile-single:
run-single:
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 3.00
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 4.00
BUILD SUCCESSFUL (total time: 1 second)
```

Pertanyaan

Berdasarkan percobaan 1, jawablah pertanyaan-pertanyaan yang terkait:

1. Di dalam `class Processor` dan `class Laptop`, terdapat method `setter` dan `getter` untuk masing-masing atributnya. Apakah gunanya `method setter` dan `getter` tersebut?
2. Di dalam `class Processor` dan `class Laptop`, masing-masing terdapat konstruktor default dan konstruktor berparameter. Bagaimanakah beda penggunaan dari kedua jenis konstruktor tersebut?
3. Perhatikan `class Laptop`, di antara 2 atribut yang dimiliki (`merk` dan `proc`), atribut manakah yang bertipe `object`?
4. Perhatikan `class Laptop`, pada baris manakah yang menunjukkan bahwa `class Laptop` memiliki relasi dengan `class Processor`?
5. Perhatikan pada `class Laptop`, Apakah guna dari sintaks `proc.info()`?
6. Pada `class MainPercobaan1`, terdapat baris kode:

```
Laptop l = new Laptop("Thinkpad", p);
```

Apakah `p` tersebut?

Dan apakah yang terjadi jika baris kode tersebut diubah menjadi:

```
Laptop l = new Laptop("Thinkpad", new Processor("Intel i5",  
3));
```

Bagaimanakah hasil program saat dijalankan, apakah ada perubahan?

Jawaban

1. Kegunaan method `setter` dan `getter` adalah untuk mengakses atribut yang di private pada class yang lain.
2. Konstruktor perparameter untuk memasukkan nilai pada saat instansiasi sedangkan konstruktor default nilai di masukkan saat inisialisasi.
3. Atribut yang bertipe object adalah `proc`
4. Pada atribut private `Processor proc`
5. Kegunaan sintaks `proc.info()` adalah untuk menampilkan info yang terdapat pada class `Processor`
6. Pada kode ini `p` adalah object

```
Laptop l = new Laptop("Thinkpad", p);
```

Dan apabila baris kode tersebut dirubah hasilnya sama saja

```
package Jobsheet4.Percobaan1;  
public class MainPercobaan1 {  
    public static void main(String args[]){  
        Processor p = new Processor("Intel i5",3);  
        Laptop L = new Laptop("Thinkpad",  
            new Processor("Intel i5",3));  
        L.info();  
    }  
}
```

Hasil :

```
run-single:  
Merk Laptop = Thinkpad  
Merk Processor = Intel i5  
Cache Memory = 3.00  
Merk Laptop = Thinkpad  
Merk Processor = Intel i5  
Cache Memory = 4.00  
BUILD SUCCESSFUL (total time: 1 second)
```

Percobaan 2

1. Class Mobil

```
5  | */
6  | package Jobsheet4.Percobaan2;
7  | public class Mobil {
8  |     private String nama;
9  |     private int biaya;
10 |
11 |     Mobil() {
12 |     }
13 |
14 |     public void setnama(String nama) {
15 |         this.nama = nama;
16 |     }
17 |
18 |     public String getnama() {
19 |         return nama;
20 |     }
21 |
22 |     public void setbiaya(int biaya) {
23 |         this.biaya = biaya;
24 |     }
25 |
26 |     public int getbiaya() {
27 |         return biaya;
28 |     }
29 |
30 |     public int hitungBiayaMobil(int hari) {
31 |         return biaya*hari;
32 |     }
33 | }
```

2. Class Sopir

```
6 package Jobsheet4.Percobaan2;
7
8 /**
9  *
10  * @author DEBORA
11  */
12 public class Sopir {
13     private String nama;
14     private int biaya;
15
16     Sopir(){
17     }
18
19     public void setnama(String nama){
20         this.nama = nama;
21     }
22
23     public String getnama(){
24         return nama;
25     }
26
27     public void setbiaya(int biaya){
28         this.biaya = biaya;
29     }
30
31     public int getbiaya(){
32         return biaya;
33     }
34
35     public int hitungBiayaMobil(int hari){
36         return biaya*hari;
37     }
38 }
```

3. Class Pelanggan

```
12 public class Pelanggan {
13     private String nama;
14     private Mobil mobil;
15     private Sopir sopir;
16     private int hari;
17
18     public void setnama(String nama){
19         this.nama = nama;
20     }
21     public String getnama(){
22         return nama;
23     }
24     public void setmobil(Mobil mobil){
25         this.mobil = mobil;
26     }
27     public Mobil getmobil(){
28         return mobil;
29     }
30     public void setsopir(Sopir sopir){
31         this.sopir = sopir;
32     }
33     public Sopir getsopir(){
34         return sopir;
35     }
36     public void sethari(int hari){
37         this.hari = hari;
38     }
39     public int gethari(){
40         return hari;
41     }
42
43     public int hitungBiayaTotal(){
44         return mobil.hitungBiayaMobil(hari)+
45             sopir.hitungBiayaMobil(hari);
46     }
```

4. Class MainPercobaan2

```
12 public class MainPercobaan2 {  
13     public static void main(String args[]){  
14         Mobil m = new Mobil();  
15         m.setmerk("Avanza");  
16         m.setbiaya(350000);  
17         Sopir s = new Sopir();  
18         s.setnama("John Doe");  
19         s.setbiaya(200000);  
20         Pelanggan p = new Pelanggan();  
21         p.setnama("Jane Doe");  
22         p.setmobil(m);  
23         p.setsopir(s);  
24         p.sethari(2);  
25         System.out.println("Biaya Total = "  
26             +p.hitungBiayaTotal());  
27     }  
28 }  
29
```

Hasil

```
run-single:  
Biaya Total = 1100000  
BUILD SUCCESSFUL (total time: 1 second)
```

Pertanyaan

1. Perhatikan *class* Pelanggan. Pada baris program manakah yang menunjukkan bahwa *class* Pelanggan memiliki relasi dengan *class* Mobil dan *class* Sopir?
2. Perhatikan *method* hitungBiayaSopir pada *class* Sopir, serta *method* hitungBiayaMobil pada *class* Mobil. Mengapa menurut Anda *method* tersebut harus memiliki argument hari?
3. Perhatikan kode dari *class* Pelanggan. Untuk apakah perintah mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari)?
4. Perhatikan *class* MainPercobaan2. Untuk apakah sintaks p.setMobil(m) dan p.setSopir(s)?
5. Perhatikan *class* MainPercobaan2. Untuk apakah proses p.hitungBiayaTotal() tersebut?
6. Perhatikan *class* MainPercobaan2, coba tambahkan pada baris terakhir dari *method* main dan amati perubahan saat di-run!

```
System.out.println(p.getMobil().getMerk());
```

Jadi untuk apakah sintaks p.getMobil().getMerk() yang ada di dalam *method* main tersebut?

Jawaban

```
private Mobil mobil;
```

```
1. private Sopir sopir;
```

2. Pada method tersebut harus memiliki argumen hari dikarenakan penghitungan biaya mobil dan sopir dikenakan tarif biaya perhari.

3. Perintah `mobil.hitungBiayaMobil(hari)` dan `sopir.hitungBiayaSopir(hari)` digunakan untuk menghitung total biaya mobil dan sopir.

4. Sintaks `p.setMobil(m)` dan `p.setSopir(s)` digunakan untuk mengeset objek mobil dan sopir.

```
run-single:
```

```
Biaya Total = 1100000
```

```
Avanza
```

```
5. BUILD SUCCESSFUL (total time: 0 seconds)
```

Percobaan 4

1. Class Pegawai

```
6 package Jobsheet4.Percobaan3;
7 public class Pegawai {
8     private String NIP;
9     private String nama;
10
11     Pegawai() {
12     }
13
14     Pegawai(String nama, String NIP) {
15         this.nama = nama;
16         this.NIP = NIP;
17     }
18
19     public void setnip(String NIP) {
20         this.NIP = NIP;
21     }
22
23     public String getnip() {
24         return NIP;
25     }
26
27     public void setnama(String nama) {
28         this.nama = nama;
29     }
30
31     public String getnama() {
32         return nama;
33     }
34
35     public String info() {
36         String info = "";
37         info += "NIP: " + this.NIP + "\n";
38         info += "Nama: " + this.nama + "\n";
39         return info;
40     }
}
```

2. Class Kereta Api

```
6 package Jobsheet4.Percobaan3;
7 public class KeretaApi {
8     private String nama;
9     private String kelas;
10    private Pegawai masinis;
11    private Pegawai asisten;
12
13    KeretaApi(String nama, String kelas, Pegawai masinis){
14        this.nama = nama;
15        this.kelas = kelas;
16        this.masinis = masinis;
17    }
18
19    KeretaApi(String nama, String kelas, Pegawai masinis, Pegawai asisten){
20        this.nama = nama;
21        this.kelas = kelas;
22        this.masinis = masinis;
23        this.asisten = asisten;
24    }
25
26    public void setnama(String nama){
27        this.nama = nama;
28    }
29    public String getnama(){
30        return nama;
31    }
32    public void setkelas(String kelas){
33        this.kelas = kelas;
34    }
35    public String getkelas(){
36        return kelas;
37    }
38    public void setmasinis(Pegawai masinis){
39        this.masinis = masinis;
40    }
41    public Pegawai getmasinis(){
42        return masinis;
43    }
44    public void setasisten(Pegawai asisten){
45        this.asisten = asisten;
46    }
47    public Pegawai getasisten(){
48        return asisten;
49    }
50    public String info(){
51        String info = "";
52        info += "Nama: "+this.nama+"\n";
53        info += "Kelas: "+this.kelas+"\n";
54        info += "Masinis: "+this.masinis+"\n";
55        info += "Asisten: "+this.asisten+"\n";
56        return info;
57    }
58 }
```

3. Class MainPercobaan3

```
6 package Jobsheet4.Percobaan3;
7
8 /**
9  *
10  * @author DEBORA
11  */
12 public class MainPercobaan3 {
13     public static void main (String args[]){
14         Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");
15         Pegawai asisten = new Pegawai("4567", "Patrick Star");
16         KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis, asisten);
17
18         System.out.println(keretaApi.info());
19     }
20 }
21
22 }
```

Output

```
run-single:
Nama: Gaya Baru
Kelas: Bisnis
Masinis: Jobsheet4.Percobaan3.Pegawai@1218025c
Asisten: Jobsheet4.Percobaan3.Pegawai@816f27d
```

```
BUILD SUCCESSFUL (total time: 0 seconds)
```

Pertanyaan

1. Di dalam *method* `info()` pada *class* `KeretaApi`, baris `this.masinis.info()` dan `this.asisten.info()` digunakan untuk apa ?
2. Buatlah *main* program baru dengan nama *class* `MainPertanyaan` pada *package* yang sama. Tambahkan kode berikut pada *method* `main()` !

```
Pegawai masinis = new Pegawai("1234", "Spongebob
Squarepants");
KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis",
masinis);
```

```
System.out.println(keretaApi.info());
```

3. Apa hasil output dari *main* program tersebut ? Mengapa hal tersebut dapat terjadi ?
4. Perbaiki *class* `KeretaApi` sehingga program dapat berjalan !

Jawaban

1. Method `info()` pada *class* `KeretaApi` dan baris `this.masinis.info()` dan `this.asisten.info()` adalah untuk memunculkan info dari objek `masinis` dan `asisten`

```

10 |      * @author DEBORA
11 |      */
12 |      public class MainPercobaan3 {
13 |          public static void main (String args[]){
14 |              Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");
15 |              Pegawai asisten = new Pegawai("4567", "Patrick Star");
16 |              KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis, asisten);
17 |
18 |              System.out.println(keretaApi.info());
19 |          }
20 |      }

```

2.

3. Hasilnya

```

run-single:
Nama: Gaya Baru
Kelas: Bisnis
Masinis: Jobsheet4.Percobaan3.Pegawai@1218025c
Asisten: null

```

BUILD SUCCESSFUL (total time: 0 seconds)

Percobaan 4

1. Class Penumpang

```

9 |      *
10 |      * @author DEBORA
11 |      */
12 |      public class Penumpang {
13 |          private String ktp;
14 |          private String nama;
15 |
16 |          Penumpang() {
17 |          }
18 |          Penumpang(String ktp, String nama){
19 |              this.ktp=ktp;
20 |              this.nama=nama;
21 |          }
22 |          public void setktp(String ktp){
23 |              this.ktp=ktp;
24 |          }
25 |          public String getktp(){
26 |              return ktp;
27 |          }
28 |          public void setnama(String nama){
29 |              this.nama=nama;
30 |          }
31 |          public String getnama(){
32 |              return nama;
33 |          }
34 |          public String info(){
35 |              String info = "";
36 |              info+="KTP: "+ktp+"\n";
37 |              info+="Nama: "+nama+"\n";
38 |              return info;
39 |          }
40 |      }

```

2. Class Kursi

```
10  * @author DEBORA
11  */
12  public class Kursi {
13      private String nomor;
14      private Penumpang penumpang;
15
16      public Kursi(String nomor){
17          this.nomor=nomor;
18      }
19      public void setnomor(String nomor){
20          this.nomor=nomor;
21      }
22      public String getnomor(){
23          return nomor;
24      }
25      public void setpenumpang(Penumpang penumpang){
26          this.penumpang=penumpang;
27      }
28      public Penumpang getpenumpang(){
29          return penumpang;
30      }
31      public String info(){
32          String info = "";
33          info+="Nomor: "+nomor+"\n";
34          if(this.penumpang != null){
35              info+="Penumpang: "+penumpang.info()+"\n";
36          }
37          return info;
38      }
39  }
40
```

3. Class Gerbong

```
12 public class Gerbong {
13     private String kode;
14     private Kursi[] arrayKursi;
15
16     public Gerbong(String kode, int jumlah){
17         this.kode=kode;
18         this.arrayKursi=new Kursi[jumlah];
19         this.initKursi();
20     }
21     private void initKursi(){
22         for (int i=0; i<arrayKursi.length; i++){
23             this.arrayKursi[i] = new Kursi(String.valueOf(i+1));
24         }
25     }
26
27     public void setKode(String kode){
28         this.kode=kode;
29     }
30     public String getkode(){
31         return kode;
32     }
33     public void setpenumpang(Penumpang penumpang, int nomor){
34         this.arrayKursi[nomor-1].setpenumpang(penumpang);
35     }
36     public void setarrayKursi(Kursi[] arrayKursi){
37         this.arrayKursi=arrayKursi;
38     }
39     public String info(){
40         String info = "";
41         info+="Kode: "+kode+"\n";
42         for(Kursi kursi:arrayKursi){
43             info+=kursi.info();
44         }
45         return info;
46     }
}
```

4. Class MainPercobaan4

```
6 package Jobsheet4.Percobaan4;
7
8 /**
9  *
10  * @author DEBORA
11  */
12 public class MainPercobaan4 {
13     public static void main(String args[]){
14         Penumpang p = new Penumpang("12345", "Mr.Crab");
15         Gerbong gerbong = new Gerbong("a",10);
16         gerbong.setpenumpang(p, 1);
17         System.out.println(gerbong.info());
18     }
19
20 }
```

Output

```
run-single:
Kode: a
Nomor: 1
Penumpang: KTP: 12345
Nama: Mr.Crab
```

```
Nomor: 2
Nomor: 3
Nomor: 4
Nomor: 5
Nomor: 6
Nomor: 7
Nomor: 8
Nomor: 9
Nomor: 10
```

BUILD SUCCESSFUL (total time: 1 second)

Pertanyaan

1. Pada *main* program dalam *class* MainPercobaan4, berapakah jumlah kursi dalam Gerbong A ?
2. Perhatikan potongan kode pada *method* *info()* dalam *class* *Kursi*. Apa maksud kode tersebut ?

```
...
if (this.penumpang != null) {
    info += "Penumpang: " + penumpang.info() + "\n";
}
...
```

3. Mengapa pada *method* *setPenumpang()* dalam *class* *Gerbong*, nilai nomor dikurangi dengan angka 1 ?
4. Instansiasi objek baru budi dengan tipe *Penumpang*, kemudian masukkan objek baru tersebut pada gerbong dengan *gerbong.setPenumpang(budi, 1)*. Apakah yang terjadi ?
5. Modifikasi program sehingga tidak diperkenankan untuk menduduki kursi yang sudah ada penumpang lain !

Jawaban

1. Jumlah kursi pada gerbong A adalah 10.
2. Arti kode tersebut adalah jika atribut penumpang tidak kosong maka info ditambah info penumpang.
3. Nilai nomor dikurangi dengan angka 1 dikarenakan array dimulai dengan angka 0.

4.

```
gerbong.setpenumpang(p, 1);
gerbong.setpenumpang(p, 2);
```

Output

run-single:

Kode: a

Nomor: 1

Penumpang: KTP: 12345

Nama: Mr.Crab

Nomor: 2

Penumpang: KTP: 12345

Nama: Mr.Crab

Nomor: 3

Nomor: 4

Nomor: 5

Nomor: 6

Nomor: 7

Nomor: 8

Nomor: 9

Nomor: 10

BUILD SUCCESSFUL (total time: 1 second)

TUGAS

Mahasiswa

-nim:String
-nama:String

+Mahasiswa(nim:int, nama:String)
+setnim(nim:String)
+getnim():String
+setnama(nama:String)
+getnama():String
+info():String

Dosen

-nip:String
-nama:String

+Dosen(nip:int, nama:String)
+setnip(nip:int)
+getnip():String
+setnama(nama:String)
+getnama():String
+info():String

MataKuliah

-kodeMatkul:String
-namaMatkul:String
-mahasiswa:Mahasiswa
-dosen:Dosen
-ruangkelas:RuangKelas

+MataKuliah(kodeMatkul:String, namaMatkul:String)
+MataKuliah(kodeMatkul:String, namaMatkul:String, mahasiswa:Mahasiswa,
dosen:Dosen, ruangkelas:RuangKelas)
+setkodeMatkul(kodeMatkul:String)
+getkodeMatkul:String
+setnamaMatkul(namaMatkul:String)
+getnamaMatkul:String
+setmahasiswa(mahasiswa:Mahasiswa)
+getmahasiswa():Mahasiswa
+setdosen(dosen:Dosen)
+getdosen():Dosen
+setruangkelas(ruangkelas:RuangKelas)
+getruangkelas():RuangKelas
+info():void

RuangKelas

- kodekelas:String

+RuangKelas(kodekelas:String)
+setkode(kodekelas:String)
+getkode():String
info():String

