



NAMA : HALDA DINI SILMA ROSIDA
NIM : 2041720014
KELAS : TI-2C
MATERI : PRAKTIKUM PBO (JOBSHEET 4)

Percobaan 1

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 ?

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. Di dalam *class* Processor dan *class* Laptop, masing-masing terdapat konstruktor default dan konstruktor berparameter. Bagaimanakah beda penggunaan dari kedua jenis konstruktor tersebut ?

Jawab :

Konstruktor default :

- Konstruktor yang tidak menggunakan argumen dikenal sebagai konstruktor default.
- Compiler memasukkan konstruktor no-arg default setelah kompilasi, jika tidak ada konstruktor eksplisit yang didefinisikan di kelas dan datamember diinisialisasi dengan 0.
- Tidak perlu melewati parameter apa pun saat membangun objek baru menggunakan konstruktor default.
- Konstruktor default digunakan untuk menginisialisasi objek dengan data yang sama.

Konstruktor parametrized :

- Konstruktor yang mengambil satu atau lebih argumen dikenal sebagai konstruktor berparameter.
- Ketika konstruktor parametrized didefinisikan di kelas, maka programmer perlu mendefinisikan konstruktor no-arg default secara eksplisit jika diperlukan.
- Setidaknya satu atau lebih parameter perlu dilewati saat membangun objek baru menggunakan konstruktor argumen.
- Sedangkan konstruktor parametrized digunakan untuk membuat objek yang berbeda dengan data yang berbeda.

3. Perhatikan *class* Laptop, di antara 2 atribut yang dimiliki (*merk* dan *proc*), atribut manakah yang bertipe *object* ?

Jawab : Atribut yang bertipe object adalah atribut *proc*.

4. Perhatikan *class* Laptop, pada baris manakah yang menunjukkan bahwa *class* Laptop memiliki relasi dengan *class* Processor ?

Jawab :

Pada baris ke-4

```
4 private Processor proc;
```

5. Perhatikan pada *class* Laptop, Apakah guna dari sintaks *proc.info()* ?

Jawab : Untuk dapat mengakses method *info* di class Processor tanpa harus mengetik ulang di method *info* di class Laptop

6. Pada *class* MainPercobaan1, terdapat baris kode:

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

Apakah *p* tersebut ?

Jawab : *p* tersebut adalah object (instansiasi) dari class Processor.

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

Jawab : Hasilnya sama tidak ada perubahan. Karena instansiasi objeknya berada langsung didalam



NAMA : HALDA DINI SILMA ROSIDA
NIM : 2041720014
KELAS : TI-2C
MATERI : PRAKTIKUM PBO (JOBSHEET 4)

parameter konstruktor laptop sehingga tidak perlu membuat variabel baru bertipe data objek.

Percobaan 2

Pertanyaan

1. Perhatikan *class* Pelanggan. Pada baris program manakah yang menunjukkan bahwa *class* Pelanggan memiliki relasi dengan *class* Mobil dan *class* Sopir ?

Jawab : pada baris ke 4-5

```
4 private Mobil mobil;  
5 private Sopir sopir;
```

2. Perhatikan *method* hitungBiayaSopir pada *class* Sopir, serta *method* hitungBiayaMobil pada *class* Mobil. Mengapa menurut Anda *method* tersebut harus memiliki argument hari ?

Jawab : Ini karena pada *class* Pelanggan terdapat atribut yang berisi hari dan atribut hari ini digunakan untuk melakukan *method* hitungBiayaTotal yang didalam *method* ini terdapat pemanggilan *method* hitungBiayaMobil dan hitungBiayaSopir. Kedua *method* ini memerlukan parameter hari untuk melakukan proses yang terjadi didalamnya yaitu mengalikan biaya dan jumlah hari

3. Perhatikan kode dari *class* Pelanggan. Untuk apakah perintah mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari) ?

Jawab : Untuk menghitung biaya mobil dan biaya sopir tanpa harus mengetik ulang di *method* hitungBiayaTotal. Bisa langsung memanggil object dari *class* Mobil dan *class* Sopir

4. Perhatikan *class* MainPercobaan2. Untuk apakah sintaks p.setMobil(m) dan p.setSopir(s) ?

Jawab : Sintaks ini digunakan untuk melakukan input data pada *class* Pelanggan, p.setMobil(m) untuk mengisi atribut mobil dengan nilai yang terdapat pada object m dan p.setSopir(s) untuk mengisi atribut sopir dengan nilai yang terdapat pada object s.

5. Perhatikan *class* MainPercobaan2. Untuk apakah proses p.hitungBiayaTotal() tersebut ?

Jawab : Untuk menghitung total biaya sewa per harinya yang harus dibayarkan.

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?

Jawab : Jadi untuk memanggil nama merek yang ada pada *class* Mobil dari data yang dimasukkan

```
run:  
Biaya Total : 1100000  
Avanza  
BUILD SUCCESSFUL (total time: 0 seconds)
```



NAMA : HALDA DINI SILMA ROSIDA
NIM : 2041720014
KELAS : TI-2C
MATERI : PRAKTIKUM PBO (JOBSHEET 4)

Percobaan 3

Pertanyaan

1. Di dalam *method* info() pada *class* KeretaApi, baris this.masinis.info() dan this.asisten.info() digunakan untuk apa ?

Jawab : Digunakan untuk memanggil *method* info() yang terdapat pada *class* masinis dan asisten

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());

Jawab :

```
1 package Main.relasiclass.percobaan3;
2
3 public class MainPertanyaan {
4
5     public static void main(String[] args) {
6
7         Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");
8         KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis);
9
10        System.out.println(keretaApi.info());
11    }
12 }
```

3. Apa hasil output dari *main* program tersebut ? Mengapa hal tersebut dapat terjadi ?

Jawab :

```
run:
Exception in thread "main" java.lang.NullPointerException
    at Main.relasiclass.percobaan3.KeretaApi.info(KeretaApi.java:60)
    at Main.relasiclass.percobaan3.MainPertanyaan.main(MainPertanyaan.java:10)
Java Result: 1
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ini terjadi karena pada *method* info() pada *class* keretaApi mengharuskan outputnya menampilkan data asisten juga. Sedangkan data asisten belum diinputkan pada *class* KeretaApi sehingga program memunculkan bahwa ada atribut yang kurang dan tidak boleh bernilai null atau kosong.

4. Perbaiki *class* KeretaApi sehingga program dapat berjalan !

Jawab :

```
62         info += "Asisten : " + this.asisten.info() + "\n";
63     }
64     return info;
```

```
run:
Nama: Gaya Baru
Kelas: Bisnis
Masinis: Nip : 1234
Nama : Spongebob Squarepants

Asisten: Nip : 4567
Nama : Patrick Star
```



NAMA : HALDA DINI SILMA ROSIDA
NIM : 2041720014
KELAS : TI-2C
MATERI : PRAKTIKUM PBO (JOBSHEET 4)

Percobaan 4

Pertanyaan

1. Pada *main* program dalam *class* MainPercobaan4, berapakah jumlah kursi dalam Gerbong A ?

Jawab : ada 10 kursi

2. Perhatikan potongan kode pada *method* *info()* dalam *class* *Kursi*. Apa maksud kode tersebut ?

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

Jawab : Kode tersebut maksudnya adalah untuk memeriksa nilai didalam atribut penumpang yang ada pada class *Kursi*. Jika atribut penumpang tidak kosong, maka tambahkan info dengan info yang didapat dari pemanggilan *info* dalam class penumpang.

3. Mengapa pada *method* *setPenumpang()* dalam *class* *Gerbong*, nilai nomor dikurangi dengan angka 1 ?

Jawab : Ini dilakukan agar array menerima indeks yang benar karena array dimaulai dari indeks ke-0 bukannya 1. Jadi ketika ingin melakukan inputan pada indeks yang benar, namun pada inputannya kita gunakan range mulai dari 1 maka nilai harus dikurangi 1

4. Instansiasi objek baru budi dengan tipe *Penumpang*, kemudian masukkan objek baru tersebut pada gerbong dengan *gerbong.setPenumpang(budi, 1)*. Apakah yang terjadi ?

Jawab :

```
penumpang Budi = new penumpang("223344", "Budi");  
gerbong.setPenumpang(Budi, 1);  
System.out.println(gerbong.info());
```

run:

Kode: A

Nomor: 1

Penumpang: Ktp: 12345

Nama: Mr. Krab

Nomor: 2

Nomor: 3

Nomor: 4

Nomor: 5

Nomor: 6

Nomor: 7

Nomor: 8

Nomor: 9

Nomor: 10

Kode: A

Nomor: 1

Penumpang: Ktp: 223344

Nama: Budi

Nomor: 2

Nomor: 3

Nomor: 4

Nomor: 5

Nomor: 6

Nomor: 7

Nomor: 8

Nomor: 9

Nomor: 10

BUILD SUCCESSFUL (total time: 0 seconds)



NAMA : HALDA DINI SILMA ROSIDA
NIM : 2041720014
KELAS : TI-2C
MATERI : PRAKTIKUM PBO (JOBSHEET 4)

5. Modifikasi program sehingga tidak diperkenankan untuk menduduki kursi yang sudah ada penumpang lain !

Jawab :

```
29 public void setPenumpang(penumpang penumpang, int nomor) {  
30     int index = nomor - 1;  
31     if(arrayKursi[index].getPenumpang() != null) {  
32         System.out.println("Kursi telah terisi, silahkan pilih kursi lain!");  
33     } else {  
34         this.arrayKursi[index].setPenumpang(penumpang);  
35     }  
36 }
```

run:

```
Kode: A  
Nomor: 1  
Nomor: 2  
Penumpang: Ktp: 12345  
Nama: Mr. Krab
```

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

Kursi telah terisi, silahkan pilih kursi lain!

```
Kode: A  
Nomor: 1  
Nomor: 2  
Penumpang: Ktp: 12345  
Nama: Mr. Krab
```

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

BUILD SUCCESSFUL (total time: 0 seconds)



NAMA : HALDA DINI SILMA ROSIDA
NIM : 2041720014
KELAS : TI-2C
MATERI : PRAKTIKUM PBO (JOBSHEET 4)

TUGAS!

Class Frozen

```
1 package Tugas;
2 public class Frozen {
3     private String nama;
4
5     Frozen() {
6
7     }
8
9     Frozen(String nama) {
10        this.nama = nama;
11    }
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 String info() {
23        String info = "";
24        info += "Nama: " + nama + "\n";
25        return info;
26    }
27 }
```

Class Keluarga

```
1 package Tugas;
2 public class Keluarga {
3     private String keluarga;
4     private Frozen ortu;
5
6     Keluarga() {
7
8     }
9
10    public void setKeluarga(String keluarga) {
11        this.keluarga = keluarga;
12    }
13
14    public String getKeluarga() {
15        return keluarga;
16    }
17
18    public void setorangTua(Frozen ortu) {
19        this.ortu = ortu;
20    }
21
22    public Frozen getorangTua() {
23        return ortu;
24    }
25
26    public String info() {
27        String info = "";
28        info += "Keluarga : " + keluarga + "\n";
29        info += "Ortu : " + ortu + "\n";
30        return info;
31    }
32 }
```

Class Teman

```
1 package Tugas;
2 public class Teman {
3     private String namaTeman;
4
5     Teman() {
6
7     }
8
9     public void setnamaTeman(String namaTeman) {
10        this.namaTeman = namaTeman;
11    }
12
13    public String getnamaTeman() {
14        return namaTeman;
15    }
16 }
```




NAMA : HALDA DINI SILMA ROSIDA
NIM : 2041720014
KELAS : TI-2C
MATERI : PRAKTIKUM PBO (JOBSHEET 4)

Class Peliharaan

```
1 package Tugas;  
2 public class Peliharaan {  
3     private String namaHewan;  
4  
5     Peliharaan() {  
6  
7     }  
8  
9     public void setnamaHewan(String namaHewan) {  
10        this.namaHewan = namaHewan;  
11    }  
12  
13    public String getnamHewan() {  
14        return namaHewan;  
15    }  
16 }
```

Class Main

```
1 package Tugas;  
2 public class Main {  
3     public static void main(String[] args) {  
4         Frozen f = new Frozen();  
5         f.setNama("Elsa");  
6         Keluarga k = new Keluarga();  
7         k.setKeluarga("Kerajaan");  
8         Teman t = new Teman();  
9         t.setnamaTeman("Kristoff");  
10        Peliharaan p = new Peliharaan();  
11        p.setnamaHewan("Sven");  
12  
13        System.out.println(f.getNama());  
14        System.out.println(k.getKeluarga());  
15        System.out.println(t.getnamaTeman());  
16        System.out.println(p.getnamHewan());  
17    }  
18 }
```

Hasil

```
run:  
Elsa  
Kerajaan  
Kristoff  
Sven  
BUILD SUCCESSFUL (total time: 0 seconds)
```