

# Assignment Paper

**OBJECT ORIENTED PROGRAMMING (OOP)  
PRACTICUM  
MOCHAMAD WILDANI AZIZI**

## Percobaan 1

- a. Buatlah class Processor dalam package tersebut

```
1  package JS4;
2
3  public class Processor {
4      private String merk;
5      private double cache;
6
7      // Default constructor
8      public Processor() {
9      }
10
11     // Constructor with parameters
12     public Processor(String merk, double cache) {
13         this.merk = merk;
14         this.cache = cache;
15     }
16
17     // Getter for 'merk'
18     public String getMerk() {
19         return merk;
20     }
21
22     // Setter for 'merk'
23     public void setMerk(String merk) {
24         this.merk = merk;
25     }
26
27     // Getter for 'cache'
28     public double getCache() {
29         return cache;
30     }
31
32     // Setter for 'cache'
33     public void setCache(double cache) {
34         this.cache = cache;
35     }
36
37     // Method to display information
38     public void info() {
39         System.out.printf("Merk Processor = %s\n", merk);
40         System.out.printf("Cache Memory = %.2f\n", cache);
41     }
42 }
43
```

- b. Kemudian buatlah class Laptop di dalam package yang telah anda buat.

```
1  package JS4;
2
3  public class Laptop {
4      private String merk;
5      private Processor proc;
6
7      // Default constructor
8      public Laptop() {
9          this.merk = "Default Laptop Merk";
10         this.proc = new Processor(); // Menggunakan constructor default dari Processor
11     }
12
13     // Constructor with parameters
14     public Laptop(String merk, Processor proc) {
15         this.merk = merk;
16         this.proc = proc;
17     }
18
19     // Method to display information
20     public void info() {
21         System.out.println("Merk Laptop = " + merk);
22         proc.info();
23     }
24
25     // Getter for 'merk'
26     public String getMerk() {
27         return merk;
28     }
29
30     // Setter for 'merk'
31     public void setMerk(String merk) {
32         this.merk = merk;
33     }
34
35     // Getter for 'proc'
36     public Processor getProc() {
37         return proc;
38     }
39
40     // Setter for 'proc'
41     public void setProc(Processor proc) {
42         this.proc = proc;
43     }
44 }
45
```

- c. Pada package yang sama, buatlah class MainPercobaan1 yang berisi method main().

```

1  package JS4;
2
3  public class MainPercobaan1 {
4      public static void main(String[] args) {
5          // Deklarasikan dan instansiasi objek Processor
6          Processor p = new Processor("Intel i5", 3);
7
8          // Deklarasikan dan instansiasi objek Laptop menggunakan objek Processor sebelumnya
9          Laptop L = new Laptop("Thinkpad", p);
10
11         // Panggil method info() dari objek L
12         L.info();
13
14         // Tambahkan baris kode
15         Processor p1 = new Processor();
16         p1.setMerk("Intel i5");
17         p1.setCache(4);
18         Laptop L1 = new Laptop();
19         L1.setMerk("Thinkpad");
20         L1.setProc(p1);
21         L1.info();
22     }
23 }
24

```

- d. Compile kemudian run class MainPercobaan1, akan didapatkan hasil seperti berikut:

```

Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 3,00
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 4,00

```

## Percobaan 1

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 Method: Digunakan untuk mendapatkan nilai dari suatu atribut. Dalam class, getter biasanya dibuat dengan nama getNamaAtribut(). Contohnya, getMerk() digunakan untuk mendapatkan nilai atribut merk pada class Laptop.
- Setter Method: Digunakan untuk mengatur nilai dari suatu atribut. Dalam class, setter biasanya dibuat dengan nama setNamaAtribut(). Contohnya, setMerk(String merk) digunakan untuk mengatur nilai atribut merk pada class Laptop.

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: Merupakan konstruktor yang tidak memiliki parameter. Digunakan untuk membuat objek dengan nilai default atau inisialisasi default. Contohnya, pada Processor dan Laptop, konstruktor default mengatur nilai default untuk atribut-atributnya.
- Konstruktor Berparameter: Merupakan konstruktor yang memiliki parameter. Digunakan untuk membuat objek dengan nilai yang telah ditentukan saat pembuatan objek. Contohnya, pada Processor dan Laptop, terdapat konstruktor berparameter untuk memberikan nilai awal saat objek dibuat.

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

**Jawab:** Pada class Laptop, atribut yang bertipe object adalah proc yang merupakan objek dari class Processor. Hal ini menunjukkan adanya hubungan atau komposisi antara Laptop dan Processor.

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

**Jawab:** Relasi antara Laptop dan Processor terjadi melalui atribut proc pada class Laptop. Dengan menggunakan komposisi, objek Laptop memiliki objek Processor sebagai salah satu atributnya. Ini dijelaskan oleh konstruktor dan setter pada class Laptop yang menerima objek Processor.

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

**Jawab:** proc.info() digunakan untuk memanggil method info() yang ada pada objek Processor yang disimpan dalam atribut proc di class Laptop. Ini memungkinkan untuk menampilkan informasi tentang processor yang terpasang pada laptop, dan ini merupakan contoh dari konsep komposisi atau relasi antar kelas.

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 ?

**Jawab:** Pada baris kode Laptop l = new Laptop("Thinkpad", p);, variabel p merupakan objek dari class Processor. Dengan kata lain, p adalah sebuah instance atau objek dari class Processor.

Jika baris kode tersebut diubah menjadi Laptop l = new Laptop("Thinkpad", new Processor("Intel i5", 3));, maka objek Processor baru akan langsung dibuat pada saat pembuatan objek Laptop. Dengan kata lain, tidak perlu membuat objek Processor terlebih dahulu sebelum membuat objek Laptop.

Hasil program saat dijalankan tidak akan berubah secara signifikan, namun perubahan tersebut membuat sintaksis lebih sederhana dan langsung menciptakan objek yang dibutuhkan tanpa menyimpannya dalam variabel terpisah. Artinya, hasilnya akan tetap sama, yaitu menciptakan laptop "Thinkpad" dengan processor "Intel i5" dan cache 3.

## Percobaan 2

- a. Buatlah class Mobil di dalam package tersebut.

```
1 package JS4;
2
3 public class Mobil {
4     private String merk;
5     private int biaya;
6
7     // Constructor default
8     public Mobil() {
9         this.merk = "DefaultMerk";
10        this.biaya = 0;
11    }
12
13    // Setter untuk atribut merk
14    public void setMerk(String merk) {
15        this.merk = merk;
16    }
17
18    // Getter untuk atribut merk
19    public String getMerk() {
20        return merk;
21    }
22
23    // Setter untuk atribut biaya
24    public void setBiaya(int biaya) {
25        this.biaya = biaya;
26    }
27
28    // Getter untuk atribut biaya
29    public int getBiaya() {
30        return biaya;
31    }
32
33    // Method untuk menghitung biaya mobil berdasarkan jumlah hari
34    public int hitungBiayaMobil(int hari) {
35        return biaya * hari;
36    }
37 }
38
```

- b. Tambahkan class Sopir dengan atribut nama tipe String dan biaya tipe int dengan akses modifier private berikut dengan constructor default.



```

1  package JS4;
2
3  public class Sopir {
4      private String nama;
5      private int biaya;
6
7      // Constructor default
8      public Sopir() {
9          this.nama = "DefaultNama";
10         this.biaya = 0;
11     }
12
13     // Setter untuk atribut nama
14     public void setNama(String nama) {
15         this.nama = nama;
16     }
17
18     // Getter untuk atribut nama
19     public String getNama() {
20         return nama;
21     }
22
23     // Setter untuk atribut biaya
24     public void setBiaya(int biaya) {
25         this.biaya = biaya;
26     }
27
28     // Getter untuk atribut biaya
29     public int getBiaya() {
30         return biaya;
31     }
32
33     // Method untuk menghitung biaya sopir berdasarkan jumlah hari
34     public int hitungBiayaSopir(int hari) {
35         return biaya * hari;
36     }
37 }
38
39

```

- c. Tambahkan class Pelanggan dengan constructor default.





```
1  package JS4;
2
3  public class Pelanggan {
4      private String nama;
5      private Mobil mobil;
6      private Sopir sopir;
7      private int hari;
8
9      // Constructor default
10     public Pelanggan() {
11         this.nama = "DefaultNama";
12         this.mobil = new Mobil();
13         this.sopir = new Sopir();
14         this.hari = 0;
15     }
16
17     // Setter untuk atribut nama
18     public void setNama(String nama) {
19         this.nama = nama;
20     }
21
22     // Getter untuk atribut nama
23     public String getNama() {
24         return nama;
25     }
26
27     // Setter untuk atribut mobil
28     public void setMobil(Mobil mobil) {
29         this.mobil = mobil;
30     }
31
32     // Getter untuk atribut mobil
33     public Mobil getMobil() {
34         return mobil;
35     }
36
37     // Setter untuk atribut sopir
38     public void setSopir(Sopir sopir) {
39         this.sopir = sopir;
40     }
41
42     // Getter untuk atribut sopir
43     public Sopir getSopir() {
44         return sopir;
45     }
46
47     // Setter untuk atribut hari
48     public void setHari(int hari) {
49         this.hari = hari;
50     }
51 }
```

```
52     // Getter untuk atribut hari
53     public int getHari() {
54         return hari;
55     }
56
57     // Method untuk menghitung biaya total
58     public int hitungBiayaTotal() {
59         return mobil.hitungBiayaMobil(hari) + sopir.hitungBiayaSopir(hari);
60     }
61 }
62
63
```

- d. Buatlah class MainPercobaan2 yang berisi method main().

```

1  package JS4;
2
3  public class MainPercobaan2 {
4      public static void main(String[] args) {
5          Mobil m = new Mobil();
6          m.setMerk("Avanza");
7          m.setBiaya(350000);
8
9          Sopir s = new Sopir();
10         s.setNama("John Doe");
11         s.setBiaya(200000);
12
13         Pelanggan p = new Pelanggan();
14         p.setNama("Zayn");
15         p.setMobil(m);
16         p.setSopir(s);
17         p.setHari(2);
18
19         System.out.println("-----Rental Mobil-----");
20         System.out.println("Nama Pelanggan = " + p.getNama());
21         System.out.println("Jumlah Hari      = " + p.getHari());
22         System.out.println("Nama Sopir      = " + s.getNama());
23         System.out.println("Biaya Sopir      = " + s.getBiaya());
24         System.out.println("Mobil          = " + m.getMerk());
25         System.out.println("Biaya Mobil     = " + m.getBiaya());
26         System.out.println("Biaya Total     = " + p.hitungBiayaTotal());
27         System.out.println("-----");
28     }
29 }
30
31

```

```

-----Rental Mobil-----
Nama Pelanggan = Zayn
Jumlah Hari      = 2
Nama Sopir      = John Doe
Biaya Sopir      = 200000
Mobil           = Avanza
Biaya Mobil     = 350000
Biaya Total     = 1100000
-----

```

### Pertanyaan

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

**Jawab:** Baris yang menunjukkan relasi antara class Pelanggan, Mobil, dan Sopir adalah:

`private Mobil mobil;` & `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:** Method `hitungBiayaSopir` dan `hitungBiayaMobil` memiliki parameter hari karena biaya sopir dan biaya mobil bergantung pada jumlah hari penyewaan. Dengan memasukkan parameter hari, kita memberikan fleksibilitas pada sistem untuk menghitung biaya berdasarkan durasi penyewaan.

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

**Jawab:** Pada class `Pelanggan`, perintah `mobil.hitungBiayaMobil(hari)` digunakan untuk menghitung biaya sewa mobil berdasarkan jumlah hari yang diinput. Sementara `sopir.hitungBiayaSopir(hari)` digunakan untuk menghitung biaya sewa sopir berdasarkan jumlah hari yang diinput.

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

**Jawab:**

- Sintaks `p.setMobil(m)` digunakan untuk mengatur objek `Mobil` (m) ke dalam atribut mobil di dalam objek `Pelanggan` (p).
- Sintaks `p.setSopir(s)` digunakan untuk mengatur objek `Sopir` (s) ke dalam atribut sopir di dalam objek `Pelanggan` (p).
- Ini memungkinkan objek `Pelanggan` untuk terkait dengan objek `Mobil` dan `Sopir`.

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

**Jawab:** Proses ini digunakan untuk menghitung biaya total penyewaan yang mencakup biaya mobil dan biaya sopir, dengan memanfaatkan method `hitungBiayaTotal` pada class `Pelanggan`.

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:** Sintaks `p.getMobil().getMerk()` digunakan untuk mendapatkan merk mobil yang sedang disewa oleh pelanggan p. Ini memberikan akses ke merk mobil yang terkait dengan objek `Pelanggan` melalui objek `Mobil`. Dengan cara ini, kita dapat menampilkan informasi tambahan atau melakukan operasi lain dengan atribut dari objek terkait.

### Percobaan 3

- a. Buatlah Class Pegawai

```
1  package JS4;
2
3  public class Pegawai {
4      private String nip;
5      private String nama;
6
7      // Constructor dengan parameter nip dan nama
8      public Pegawai(String nip, String nama) {
9          this.nip = nip;
10         this.nama = nama;
11     }
12
13     // Setter untuk atribut nip
14     public void setNip(String nip) {
15         this.nip = nip;
16     }
17
18     // Getter untuk atribut nip
19     public String getNip() {
20         return nip;
21     }
22
23     // Setter untuk atribut nama
24     public void setNama(String nama) {
25         this.nama = nama;
26     }
27
28     // Getter untuk atribut nama
29     public String getNama() {
30         return nama;
31     }
32
33     // Method info untuk mendapatkan informasi pegawai
34     public String info() {
35         String info = "";
36         info += "Nip: " + this.nip + "\n";
37         info += "Nama: " + this.nama + "\n";
38         return info;
39     }
40 }
41
```

- b. Buatlah Class KeretaApi

```

1 package J54;
2
3 public class KeretaApi {
4     private String nama;
5     private String kelas;
6     private Pegawai masinis;
7     private Pegawai asisten;
8
9     // Constructor 3 parameter (nama, kelas, masinis)
10    public KeretaApi(String nama, String kelas, Pegawai masinis) {
11        this.nama = nama;
12        this.kelas = kelas;
13        this.masinis = masinis;
14    }
15
16    // Constructor 4 parameter (nama, kelas, masinis, asisten)
17    public KeretaApi(String nama, String kelas, Pegawai masinis, Pegawai asisten) {
18        this.nama = nama;
19        this.kelas = kelas;
20        this.masinis = masinis;
21        this.asisten = asisten;
22    }
23
24    // Setter untuk atribut nama
25    public void setName(String nama) {
26        this.nama = nama;
27    }
28
29    // Getter untuk atribut nama
30    public String getName() {
31        return nama;
32    }
33
34    // Setter untuk atribut kelas
35    public void setKelas(String kelas) {
36        this.kelas = kelas;
37    }
38
39    // Getter untuk atribut kelas
40    public String getKelas() {
41        return kelas;
42    }
43
44    // Setter untuk atribut masinis
45    public void setMasinis(Pegawai masinis) {
46        this.masinis = masinis;
47    }
48
49    // Getter untuk atribut masinis
50    public Pegawai getMasinis() {
51        return masinis;
52    }
53
54    // Setter untuk atribut asisten
55    public void setAsisten(Pegawai asisten) {
56        this.asisten = asisten;
57    }
58
59    // Getter untuk atribut asisten
60    public Pegawai getAsisten() {
61        return asisten;
62    }
63
64    // Method info untuk mendapatkan informasi kereta
65    public String info() {
66        String info = "";
67        info += "Nama: " + this.nama + "\n";
68        info += "Kelas: " + this.kelas + "\n";
69        info += "Masinis: " + this.masinis.info() + "\n";
70        if (this.asisten != null) {
71            info += "Asisten: " + this.asisten.info() + "\n";
72        } else {
73            info += "Asisten: -\n";
74        }
75        return info;
76    }
77 }
78
79

```

- c. Buatlah sebuah class MainPercobaan3

```
1 package JS4;
2
3 public class MainPercobaan3 {
4     public static void main(String[] args) {
5         Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");
6         Pegawai asisten = new Pegawai("4567", "Patrick Star");
7
8         KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis, asisten);
9
10        System.out.println(keretaApi.info());
11    }
12 }
13
14
```

- d. Run program

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

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

## Pertanyaan

1. Di dalam method info() pada class KeretaApi, baris this.masinis.info() dan this.asisten.info() digunakan untuk apa ?  
**Jawab:** Di dalam method info() pada class KeretaApi, baris this.masinis.info() dan this.asisten.info() digunakan untuk mendapatkan informasi dari objek Pegawai yang menjadi masinis dan asisten pada kereta. Ini memanfaatkan method info() yang ada pada class Pegawai.
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());

```

1  package JS4;
2
3  public class MainPertanyaan {
4      public static void main(String[] args) {
5          // Create a Pegawai (masinis) instance
6          Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");
7
8          // Create a KeretaApi instance with only masinis (without asisten)
9          KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis);
10
11         // Print information about the train
12         System.out.println(keretaApi.info());
13     }
14 }
15

```

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

**Jawab:**

Output yang diharapkan mungkin melibatkan sebuah kesalahan terkait atribut asisten yang bernilai null (tidak diisi). Method info() pada class KeretaApi mencakup pengecekan kondisional, tetapi dalam kasus ini, nampaknya atribut asisten tidak diatur secara eksplisit, yang dapat menyebabkan NullPointerException saat mencoba mengakses informasinya.

4. Perbaiki class KeretaApi sehingga program dapat berjalan !

**Jawab:**

```

64  // Method info untuk mendapatkan informasi kereta
65  public String info() {
66      String info = "";
67      info += "Nama: " + this.nama + "\n";
68      info += "Kelas: " + this.kelas + "\n";
69      info += "Masinis: " + (this.masinis != null ? this.masinis.info() : "-") + "\n";
70      info += "Asisten: " + (this.asisten != null ? this.asisten.info() : "-") + "\n";
71      return info;
72  }
73  }
74
75

```

```

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

Asisten: -

```



## Percobaan 4

- a. Buatlah class Penumpang, Kursi, dan Gerbong

```
1  package JS4;
2
3  public class Penumpang {
4      String ktp;
5      String nama;
6
7      Penumpang (String ktp, String nama){
8          this.ktp = ktp;
9          this.nama = nama;
10     }
11
12
13     // Setter untuk atribut kelas
14     public void setKtp(String ktp) {
15         this.ktp = ktp;
16     }
17
18     // Getter untuk atribut kelas
19     public String getKtp() {
20         return ktp;
21     }
22
23     // Setter untuk atribut nama
24     public void setNama(String nama) {
25         this.nama = nama;
26     }
27
28     // Getter untuk atribut nama
29     public String getNama() {
30         return nama;
31     }
32
33     public String info() {
34         String info = "";
35         info += "Ktp: " + ktp + "\n";
36         info += "Nama: " + nama + "\n";
37         return info;
38     }
39 }
40
```



```
1  package JS4;
2
3  public class Kursi {
4      String nomor;
5      Penumpang penumpang;
6
7      public Kursi(String valueOf) {
8      }
9
10     public void Penumpang (String nomor){
11         this.nomor = nomor;
12     }
13
14     // Setter untuk atribut nomor
15     public void setNomor(String nomor) {
16         this.nomor = nomor;
17     }
18
19     // Getter untuk atribut nomor
20     public String getNomor() {
21         return nomor;
22     }
23
24     // Setter untuk atribut prnumpang
25     public void setPenumpang(Penumpang penumpang) {
26         this.penumpang = penumpang;
27     }
28
29     // Getter untuk atribut penumpang
30     public Penumpang getPenumpang() {
31         return penumpang;
32     }
33
34     public String info() {
35         String info = "";
36         info += "Nomor: " + nomor + "\n";
37         if (this.penumpang != null) {
38             info += "Penumpang: " + penumpang.info() + "\n";
39         }
40         return info;
41     }
42 }
43
```

```

1  package JS4;
2
3  public class Gerbong {
4      private String kode;
5      private Kursi[] arrayKursi;
6
7      public Gerbong(String kode, int jumlah) {
8          this.kode = kode;
9          this.arrayKursi = new Kursi[jumlah];
10         this.initKursi();
11     }
12
13     private void initKursi() {
14         for (int i = 0; i < arrayKursi.length; i++) {
15             this.arrayKursi[i] = new Kursi(String.valueOf(i + 1));
16         }
17     }
18
19     public String info() {
20         String info = "";
21         info += "Kode: " + kode + "\n";
22         for (Kursi kursi : arrayKursi) {
23             info += kursi.info();
24         }
25         return info;
26     }
27
28     public void setPenumpang(Penumpang penumpang, int nomor) {
29         this.arrayKursi[nomor - 1].setPenumpang(penumpang);
30     }
31 }
32

```

b. Buatlah class MainPercobaan4

```

1  package JS4;
2
3  public class MainPercobaan4 {
4      public static void main(String[] args) {
5          Penumpang p = new Penumpang("12345", "Mr. Krab");
6          Gerbong gerbong = new Gerbong("A", 10);
7          gerbong.setPenumpang(p, 1);
8          System.out.println(gerbong.info());
9      }
10 }
11

```

c. Run programnya

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

Nomor: null
Nomor: null
Nomor: null
Nomor: null
Nomor: null
Nomor: null
Nomor: null
Nomor: null
Nomor: null
```

## Pertanyaan

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

**Jawab:** Jumlah kursi dalam Gerbong A adalah 10. Ini dapat dilihat dari parameter yang diteruskan saat membuat objek Gerbong pada main program: `new Gerbong("A", 10);`.

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

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

**Jawab:** Kode pada method `info()` dalam class `Kursi` memiliki maksud untuk mengecek apakah ada penumpang di kursi tersebut sebelum menambahkan informasi ke string `info`. Jika penumpang tidak null (ada penumpang), maka informasi tentang penumpang tersebut ditambahkan ke string `info`.

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

**Jawab:** Pada method `setPenumpang()` dalam class `Gerbong`, nilai nomor dikurangi 1 karena array dalam pemrograman Java (dan kebanyakan bahasa pemrograman lainnya) diindeks mulai dari 0. Jadi, jika ingin mengakses elemen pertama dalam array, kita menggunakan indeks 0. Oleh karena itu, nomor kursi yang diterima sebagai parameter dikurangi 1 agar sesuai dengan indeks array.

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

**Jawab:** Jika mencoba menggunakan `gerbong.setPenumpang(budi, 1)`, maka objek penumpang Budi akan ditempatkan di kursi pertama (indeks 0) dalam Gerbong A. Ini akan menggantikan penumpang sebelumnya jika ada.

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

**Jawab:**

```
1  package JS4;
2
3  public class Gerbong {
4      private String kode;
5      private Kursi[] arrayKursi;
6
7      public Gerbong(String kode, int jumlah) {
8          this.kode = kode;
9          this.arrayKursi = new Kursi[jumlah];
10         this.initKursi();
11     }
12
13     private void initKursi() {
14         for (int i = 0; i < arrayKursi.length; i++) {
15             this.arrayKursi[i] = new Kursi(String.valueOf(i + 1));
16         }
17     }
18
19     public String info() {
20         String info = "";
21         info += "Kode: " + kode + "\n";
22         for (Kursi kursi : arrayKursi) {
23             info += kursi.info();
24         }
25         return info;
26     }
27
28     public void setPenumpang(Penumpang penumpang, int nomor) {
29         if (this.arrayKursi[nomor - 1].getPenumpang() == null) {
30             this.arrayKursi[nomor - 1].setPenumpang(penumpang);
31         } else {
32             System.out.println("Kursi " + nomor + " sudah terisi.");
33         }
34     }
35 }
36
```

## **Tugas**

Berdasarkan latihan di pertemuan teori, rancang dengan class diagram, kemudian implementasikan ke dalam program! Studi kasus harus mewakili relasi class dari percobaan-percobaan yang telah dilakukan pada materi ini, setidaknya melibatkan minimal 4 class (class yang berisi main tidak dihitung)

**Jawab:**

```
1  package JS4;
2
3  public class Buku {
4      private int id;
5      private String judul;
6      private String pengarang;
7      private int tahunTerbit;
8
9      public Buku(int id, String judul, String pengarang, int tahunTerbit) {
10         this.id = id;
11         this.judul = judul;
12         this.pengarang = pengarang;
13         this.tahunTerbit = tahunTerbit;
14     }
15
16     // Getter dan Setter
17     public int getId() {
18         return id;
19     }
20
21     public String getJudul() {
22         return judul;
23     }
24
25     public void setJudul(String judul) {
26         this.judul = judul;
27     }
28
29     public String getPengarang() {
30         return pengarang;
31     }
32
33     public void setPengarang(String pengarang) {
34         this.pengarang = pengarang;
35     }
36
37     public int getTahunTerbit() {
38         return tahunTerbit;
39     }
40
41     public void setTahunTerbit(int tahunTerbit) {
42         this.tahunTerbit = tahunTerbit;
43     }
44 }
45
```



```
1  package JS4;
2
3  public class Anggota {
4      private int id;
5      private String nama;
6
7      public Anggota(int id, String nama) {
8          this.id = id;
9          this.nama = nama;
10     }
11
12     // Getter dan Setter
13     public int getId() {
14         return id;
15     }
16
17     public String getNama() {
18         return nama;
19     }
20
21     public void setNama(String nama) {
22         this.nama = nama;
23     }
24 }
25
26
```



```
1 package JS4;
2 import java.util.Date;
3
4 public class Peminjaman {
5     private int id;
6     private Buku buku;
7     private Anggota anggota;
8     private Date tanggalPinjam;
9     private Date tanggalKembali;
10
11     public Peminjaman(int id, Buku buku, Anggota anggota, Date tanggalPinjam, Date tanggalKembali) {
12         this.id = id;
13         this.buku = buku;
14         this.anggota = anggota;
15         this.tanggalPinjam = tanggalPinjam;
16         this.tanggalKembali = tanggalKembali;
17     }
18
19     // Getter dan Setter
20     public int getId() {
21         return id;
22     }
23
24     public Buku getBuku() {
25         return buku;
26     }
27
28     public void setBuku(Buku buku) {
29         this.buku = buku;
30     }
31
32     public Anggota getAnggota() {
33         return anggota;
34     }
35
36     public void setAnggota(Anggota anggota) {
37         this.anggota = anggota;
38     }
39
40     public Date getTanggalPinjam() {
41         return tanggalPinjam;
42     }
43
44     public void setTanggalPinjam(Date tanggalPinjam) {
45         this.tanggalPinjam = tanggalPinjam;
46     }
47
48     public Date getTanggalKembali() {
49         return tanggalKembali;
50     }
51
52     public void setTanggalKembali(Date tanggalKembali) {
53         this.tanggalKembali = tanggalKembali;
54     }
55 }
56
```

```

1  package JS4;
2
3  public class Petugas {
4      private int id;
5      private String nama;
6
7      public Petugas(int id, String nama) {
8          this.id = id;
9          this.nama = nama;
10     }
11
12     // Getter dan Setter
13     public int getId() {
14         return id;
15     }
16
17     public String getNama() {
18         return nama;
19     }
20
21     public void setNama(String nama) {
22         this.nama = nama;
23     }
24 }
25

```

```

1  package JS4;
2  import java.util.ArrayList;
3  import java.util.Date;
4
5  public class Perpustakaan {
6      private ArrayList<Buku> daftarBuku;
7      private ArrayList<Anggota> daftarAnggota;
8      private ArrayList<Peminjaman> daftarPeminjaman;
9      private Petugas petugas;
10
11     public Perpustakaan(Petugas petugas) {
12         this.petugas = petugas;
13         this.daftarBuku = new ArrayList<>();
14         this.daftarAnggota = new ArrayList<>();
15         this.daftarPeminjaman = new ArrayList<>();
16     }
17
18     // Metode untuk menambah buku, anggota, dan peminjaman
19     public void tambahBuku(Buku buku) {
20         daftarBuku.add(buku);
21     }
22
23     public void tambahAnggota(Anggota anggota) {
24         daftarAnggota.add(anggota);
25     }
26
27     public void pinjamBuku(Buku buku, Anggota anggota, Date tanggalPinjam, Date tanggalKembali) {
28         Peminjaman peminjaman = new Peminjaman(daftarPeminjaman.size() + 1, buku, anggota, tanggalPinjam, tanggalKembali);
29         daftarPeminjaman.add(peminjaman);
30     }
31
32     // Getter untuk daftarPeminjaman
33     public ArrayList<Peminjaman> getDaftarPeminjaman() {
34         return daftarPeminjaman;
35     }
36 }
37

```

```

1  package JS4;
2
3  import java.util.Date;
4
5  public class MainPerpustakaan {
6      public static void main(String[] args) {
7          // Membuat objek Petugas
8          Petugas petugas = new Petugas(1, "Petugas Perpustakaan");
9
10         // Membuat objek Perpustakaan
11         Perpustakaan perpustakaan = new Perpustakaan(petugas);
12
13         // Menambahkan beberapa buku dan anggota
14         Buku buku1 = new Buku(1, "Judul Buku 1", "Pengarang 1", 2000);
15         Buku buku2 = new Buku(2, "Judul Buku 2", "Pengarang 2", 2010);
16
17         Anggota anggota1 = new Anggota(1, "Anggota 1");
18         Anggota anggota2 = new Anggota(2, "Anggota 2");
19
20         perpustakaan.tambahBuku(buku1);
21         perpustakaan.tambahBuku(buku2);
22
23         perpustakaan.tambahAnggota(anggota1);
24         perpustakaan.tambahAnggota(anggota2);
25
26         // Melakukan peminjaman buku oleh anggota
27         Date tanggalPinjam = new Date();
28         Date tanggalKembali = new Date(tanggalPinjam.getTime() + (7 * 24 * 60 * 60 * 1000)); // Tambah 7 hari
29
30         perpustakaan.pinjamBuku(buku1, anggota1, tanggalPinjam, tanggalKembali);
31
32         // Menampilkan informasi peminjaman
33         System.out.println("Informasi Peminjaman:");
34         for (Peminjaman p : perpustakaan.getDaftarPeminjaman()) {
35             System.out.println("ID Peminjaman: " + p.getId());
36             System.out.println("Buku: " + p.getBuku().getJudul());
37             System.out.println("Anggota: " + p.getAnggota().getNama());
38             System.out.println("Tanggal Pinjam: " + p.getTanggalPinjam());
39             System.out.println("Tanggal Kembali: " + p.getTanggalKembali());
40             System.out.println("-----");
41         }
42     }
43 }
44

```

```

Informasi Peminjaman:
ID Peminjaman: 1
Buku: Judul Buku 1
Anggota: Anggota 1
Tanggal Pinjam: Tue Dec 12 02:02:14 WIB 2023
Tanggal Kembali: Tue Dec 19 02:02:14 WIB 2023
-----

```