

**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
**JOBSHEET-4: RELASI KELAS**  
TI 2C/20/SHINE DEVI OKTAVIANA R S P

**JAWABAN PERTANYAAN PERCOBAAN 1**

1. Di dalam *class* Processor dan *class* Laptop , terdapat method *setter* dan *getter* untuk masing-masing atributnya. Apakah gunanya *method setter* dan *getter* tersebut ?
  - Method setter: untuk menginputkan nama merk dan processor
  - Method getter: untuk mendapat value dari atribut merk dan proc
2. Di dalam *class* Processor dan *class* Laptop, masing-masing terdapat konstruktor default dan konstruktor berparameter. Bagaimanakah beda penggunaan dari kedua jenis konstruktor tersebut ?
  - Konstruktor default digunakan ketika atribut-atribut dari class tidak perlu memiliki nilai awal
  - Konstruktor berparameter digunakan ketika atribut-atribut dari class membutuhkan nilai awal
3. Perhatikan *class* Laptop, di antara 2 atribut yang dimiliki (*merk* dan *proc*), atribut manakah yang bertipe *object* ?
  - *proc*, karena memiliki tipe data Processor yang merupakan nama dari class
4. Perhatikan *class* Laptop, pada baris manakah yang menunjukkan bahwa *class* Laptop memiliki relasi dengan *class* Processor ?
  - Baris ke-14, 17, 28, 34

```
12 public class Laptop{
13     private String merk;
14     private Processor proc;
15     Laptop(){
16     }
17     Laptop(String merk, Processor proc){
18         this.merk=merk;
19         this.proc=proc;
20     }
21     public void info(){
22         System.out.println("Merk Laptop = "+merk);
23         proc.info();
24     }
25     public void setMerk(String merk){
26         this.merk=merk;
27     }
28     public void setProc(Processor proc){
29         this.proc=proc;
30     }
31     public String getMerk() {
32         return merk;
33     }
34     public Processor getProc() {
35         return proc;
36     }
```

**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
**JOBSHEET-4: RELASI KELAS**  
TI 2C/20/SHINE DEVI OKTAVIANA R S P

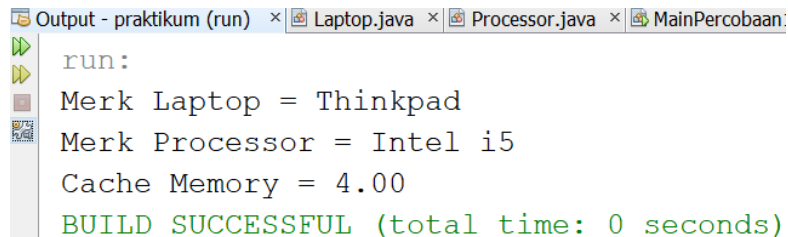
5. Perhatikan pada *class* Laptop , Apakah guna dari sintaks `proc.info()` ?
  - Untuk menampilkan method `info` dari class `Processor` yang berfungsi untuk menampilkan output berupa nama merk dan besar cache processor
6. Pada *class* `MainPercobaan1`, terdapat baris kode:  
`Laptop l = new Laptop("Thinkpad", p);`  
Apakah `p` tersebut ?
  - Object dari class `Processor` dengan konstruktor berparameter

Dan apakah yang terjadi jika baris kode tersebut diubah menjadi:

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

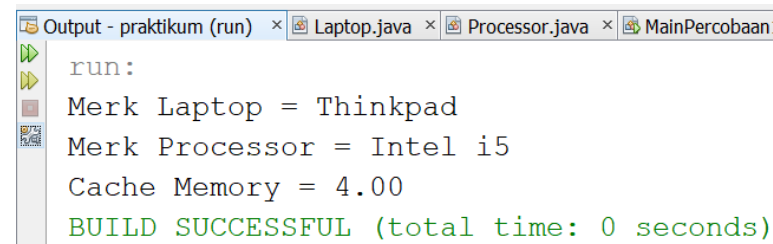
Bagaimanakah hasil program saat dijalankan, apakah ada perubahan ?

- Sebelum diubah



```
run:
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 4.00
BUILD SUCCESSFUL (total time: 0 seconds)
```

- Sesudah diubah



```
run:
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 4.00
BUILD SUCCESSFUL (total time: 0 seconds)
```

- Tidak ada perubahan karena konstruktor dari `Laptop` adalah String merk dan `Processor proc`, yang mana `proc` merupakan atribut object dari class `Processor` yang memiliki konstruktor berparameter merk dan cache. Sehingga ketika deklarasi value atribut melalui instansiasi objek `Laptop` berparameter, pada `Processor proc` bisa kita instansiasikan value atribut objek dari class `Processor`.

**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
**JOBSHEET-4: RELASI KELAS**  
TI 2C/20/SHINE DEVI OKTAVIANA R S P

**JAWABAN PERTANYAAN PERCOBAAN 2**

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

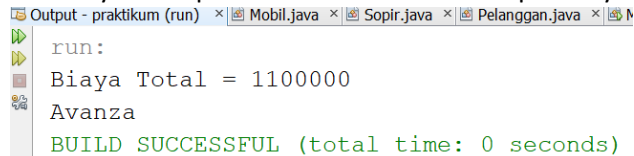
- Baris ke-14, 15, 25, 28, 31, 34, 37

```
12 public class Pelanggan {
13     private String nama;
14     private Mobil mobil;
15     private Sopir sopir;
16     private int hari;
17     Pelanggan() {
18     }
19     public void setNama(String nama) {
20         this.nama=nama;
21     }
22     public String getNama() {
23         return nama;
24     }
25     public void setMobil(Mobil mobil) {
26         this.mobil=mobil;
27     }
28     public Mobil getMobil() {
29         return mobil;
30     }
31     public void setSopir(Sopir sopir) {
32         this.sopir=sopir;
33     }
34     public Sopir getSopir() {
35         return sopir;
36     }
37     public int hitungBiayaTotal() {
38         return mobil.hitungBiayaMobil(hari)+ sopir.hitungBiayaSopir(hari);
39     }
40 }
```

2. Perhatikan *method* hitungBiayaSopir pada class Sopir, serta method hitungBiayaMobil pada class Mobil. Mengapa menurut Anda *method* tersebut harus memiliki argument hari ?
- Agar bisa menginputkan banyaknya hari, karena banyaknya hari tidak selalu sama.
3. Untuk apakah perintah mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari)?
- Agar hasil perhitungan biaya mobil dan biaya sopir dapat dijumlah menjadi biaya total
4. Perhatikan *class* MainPercobaan2. Untuk apakah sintaks p.setMobil(m) dan p.setSopir(s)?
- p.setMobil(m) digunakan untuk memasukkan value objek m dari class Mobil ke dalam method setMobil dari class Pelanggan
- p.setSopir(s) digunakan untuk memasukkan value objek s dari class Sopir ke dalam method setSopir dari class Pelanggan

**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
**JOBSHEET-4: RELASI KELAS**  
TI 2C/20/SHINE DEVI OKTAVIANA R S P

5. Untuk apakah proses `p.hitungBiayaTotal()` tersebut ?  
- Menjumlahkan biaya mobil dan biaya sopir
6. Perhatikan class `MainPercobaan2`, coba tambahkan pada baris terakhir dari *method* *main* dan amati perubahan saat di-run!  
`System.out.println(p.getMobil().getMerk());`  
- Hasilnya terdapat merk mobil setelah output biaya total



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

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

- Untuk mendapatkan value dari atribut merk dari class `Mobil` yang mana *method* tersebut berada di class `Pelanggan`

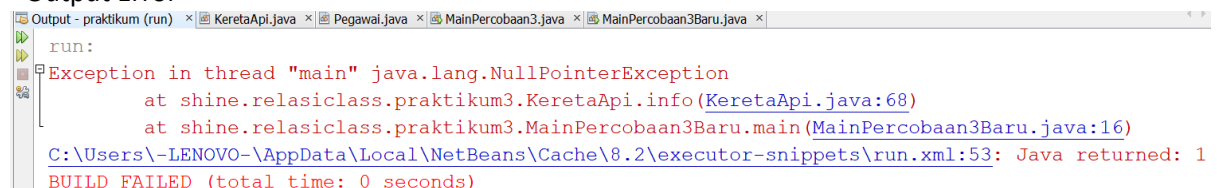
### JAWABAN PERTANYAAN PERCOBAAN 3

1. Di dalam *method* `info()` pada class `KeretaApi`, baris `this.masinis.info()` dan `this.asisten.info()` digunakan untuk apa ?  
- `this.masinis.info()` digunakan untuk memanggil *method* `info` pada class `Pegawai` melalui atribut objek `masinis`  
- `this.asisten.info()` digunakan untuk memanggil *method* `info` pada class `Pegawai` melalui atribut objek `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());
```

- Output Error



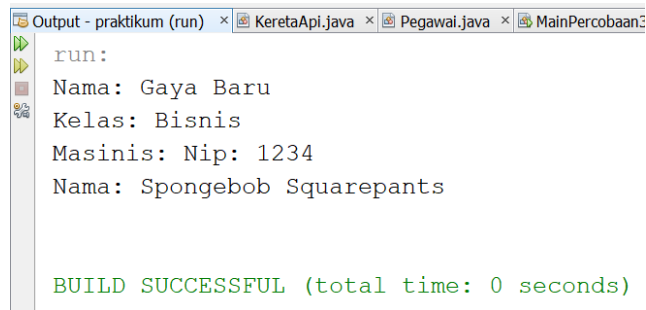
```
run:
Exception in thread "main" java.lang.NullPointerException
    at shine.relasiclass.praktikum3.KeretaApi.info(KeretaApi.java:68)
    at shine.relasiclass.praktikum3.MainPercobaan3Baru.main(MainPercobaan3Baru.java:16)
C:\Users\~\AppData\Local\NetBeans\Cache\8.2\executor-snippets\run.xml:53: Java returned: 1
BUILD FAILED (total time: 0 seconds)
```

**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
**JOBSHEET-4: RELASI KELAS**  
TI 2C/20/SHINE DEVI OKTAVIANA R S P

3. Apa hasil output dari *main* program tersebut ? Mengapa hal tersebut dapat terjadi ?
- Output error. Karena menggunakan konstruktor yang berparameter 3 yang terdiri dari nama, kelas, masinis, sedangkan di dalam method info pada class KeretaApi ada method info untuk atribut objek asisten, dan asisten tidak termasuk dalam parameter.
4. Perbaiki *class* KeretaApi sehingga program dapat berjalan !
- Membuat method info baru khusus untuk konstruktor berparameter 3 pada class KeretaApi

```
public String info() {  
    String info="";  
    info+="Nama: "+this.nama+"\n";  
    info+="Kelas: "+this.kelas+"\n";  
    info+="Masinis: "+this.masinis.info()+"\n";  
    return info;  
}
```

- Run kembali program



```
Output - praktikum (run) x KeretaApi.java x Pegawai.java x MainPercobaan3  
run:  
Nama: Gaya Baru  
Kelas: Bisnis  
Masinis: Nip: 1234  
Nama: Spongebob Squarepants  
  
BUILD SUCCESSFUL (total time: 0 seconds)
```

**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
**JOBSHEET-4: RELASI KELAS**  
TI 2C/20/SHINE DEVI OKTAVIANA R S P

**JAWABAN PERTANYAAN PRAKTIKUM 4**

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

- 10 kursi

```
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

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

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

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

- Jika kursi tidak ada penumpang, maka method akan menambahkan data info penumpang

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

- Karena jika tidak dikurangi 1, kursi yang dipakai untuk 1 penumpang akan jadi 2

```
Output - praktikum (run) x Gerbong.java x Kursi.java x MainPercobaan4.jav
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

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

**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
**JOBSHEET-4: RELASI KELAS**  
TI 2C/20/SHINE DEVI OKTAVIANA R S P

4. Instansiasi objek baru budi dengan tipe Penumpang, kemudian masukkan objek baru tersebut pada gerbong dengan `gerbong.setPenumpang(budi, 1)`. Apakah yang terjadi ?  
- Pada kursi nomor 1 yang semula diduduki oleh Mr. Krab, menjadi diduduki oleh Budi

```
public class MainPercobaan4 {  
    public static void main(String[] args) {  
        Penumpang p=new Penumpang("12345", "Mr. Krab");  
        Gerbong gerbong=new Gerbong("A", 10);  
        gerbong.setPenumpang(p, 1);  
        Penumpang budi=new Penumpang("12346", "Budi");  
        gerbong.setPenumpang(budi, 1);  
        System.out.println(gerbong.info());  
    }  
}
```

Output - praktikum (run) x Gerbong.java x Kursi.java x MainPercobaan4.java

run:  
Kode: A  
Nomor: 1  
Penumpang: Ktp: 12346  
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)

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

```
public void setPenumpang(Penumpang penumpang, int nomor){  
    this.arrayKursi[nomor-1].setPenumpang(penumpang);  
    if (this.arrayKursi[nomor-1].getPenumpang() != null) {  
        System.out.println("Sudah ditempati");  
    }  
}
```

run:  
Sudah ditempati  
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

**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
**JOBSHEET-4: RELASI KELAS**  
 TI 2C/20/SHINE DEVI OKTAVIANA R S P

**TUGAS**

```

public class TokohUtama {
    private String nama, kegiatan;
    Kelas kls;
    Sekolah sklh;
    Klub klb;

    TokohUtama(String nama, String kegiatan){
        this.nama=nama;
        this.kegiatan=kegiatan;
        this.kls=kls;
        this.sklh=sklh;
        this.klb=klb;
    }

    public void setKls(Kelas kls) {
        this.kls = kls;
    }

    public void setSklh(Sekolah sklh) {
        this.sklh = sklh;
    }

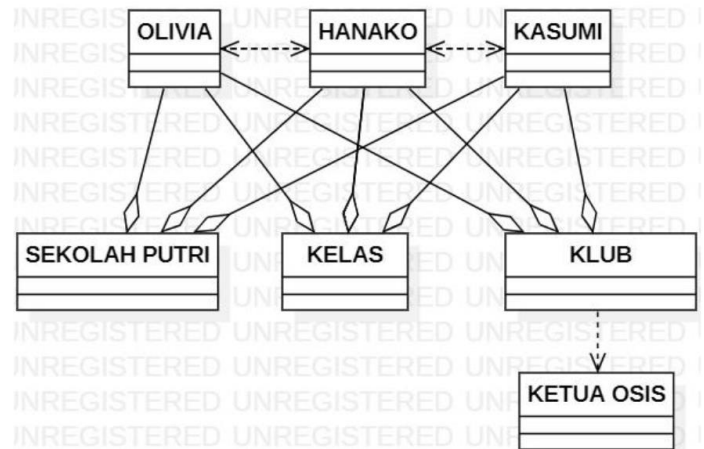
    public void setKlb(Klub klb) {
        this.klb = klb;
    }

    public Kelas getKls() {
        return kls;
    }

    public Sekolah getSklh() {
        return sklh;
    }

    public Klub getKlb() {
        return klb;
    }

    public void info(){
        System.out.println("Nama: "+nama);
        System.out.println("Kegiatan: "+kegiatan);
        System.out.println("Kelas: "+kls.getKls());
        System.out.println("Sekolah: "+sklh.getJenis());
        System.out.println("Klub: "+klb.getNamaKlub());
        klb.info();
    }
}
    
```



```

public class Sekolah {
    private String jenis;

    public void setJenis(String jenis) {
        this.jenis = jenis;
    }

    public String getJenis() {
        return jenis;
    }
}

public class Kelas {
    private String namaKls;

    public void setKls(String kls) {
        this.namaKls = kls;
    }

    public String getKls() {
        return namaKls;
    }
}
    
```



**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
**JOBSHEET-4: RELASI KELAS**  
TI 2C/20/SHINE DEVI OKTAVIANA R S P

```
public class Klub {
    private String namaKlub, kegiatan;
    KetuaOsis keputusan;

    public void setNamaKlub(String namaKlub) {
        this.namaKlub = namaKlub;
    }

    public void setKegiatan(String kegiatan) {
        this.kegiatan = kegiatan;
    }

    public void setKeputusan(KetuaOsis keputusan) {
        this.keputusan = keputusan;
    }

    public String getNamaKlub() {
        return namaKlub;
    }

    public String getKegiatan() {
        return kegiatan;
    }

    public KetuaOsis getKeputusan() {
        return keputusan;
    }

    public void info(){
        System.out.println("Kegiatan: "+kegiatan);
        keputusan.info();
    }
}

public class MainCerita {
    public static void main(String[] args) {
        Sekolah s=new Sekolah();
        s.setJenis("Sekolah Putri");
        KetuaOsis ket=new KetuaOsis("Tidak resmi");
        ket.getKeputusan();
        Klub kl=new Klub();
        kl.setNamaKlub("Klub bermain");
        kl.setKegiatan("Random");
        kl.setKeputusan(ket);
        kl.getKeputusan();
        Kelas kls=new Kelas();
        kls.setKls("2");
        TokohUtama hanako=new TokohUtama("Hanako", "Menantang bermain Kasumi & Olivia");
        hanako.setKlb(kl);
        hanako.setKls(kls);
        hanako.setSklh(s);
        hanako.getKlb();
        hanako.info();
        System.out.println("");
        TokohUtama olivia=new TokohUtama("Olivia", "Mengajak bermain & menggoda Hanako");
        olivia.setKlb(kl);
        olivia.setKls(kls);
        olivia.setSklh(s);
        olivia.getKlb();
        olivia.info();
        System.out.println("");
        TokohUtama kasumi=new TokohUtama("Kasumi", "Belajar bahasa inggris & menerima tantangan bermain");
        kasumi.setKlb(kl);
        kasumi.setKls(kls);
        kasumi.setSklh(s);
        kasumi.getKlb();
        kasumi.info();
        System.out.println("");
    }
}
```

```
public class KetuaOsis {
    private String keputusan;
    KetuaOsis(String keputusan){
        this.keputusan=keputusan;
    }

    public void setKep(String keputusan) {
        this.keputusan = keputusan;
    }

    public String getKeputusan() {
        return keputusan;
    }

    public void info(){
        System.out.println("Pemberian status klub oleh OSIS: "+keputusan);
    }
}
```

Output - Tugas4 (run) x MainCerita.java x KetuaOsis.java x Kelas.java x Klub.java x TokohUtama.java x Seku

Nama: Hanako  
Kegiatan: Menantang bermain Kasumi & Olivia  
Kelas: 2  
Sekolah: Sekolah Putri  
Klub: Klub bermain  
Kegiatan: Random  
Pemberian status klub oleh OSIS: Tidak resmi

Nama: Olivia  
Kegiatan: Mengajak bermain & menggoda Hanako  
Kelas: 2  
Sekolah: Sekolah Putri  
Klub: Klub bermain  
Kegiatan: Random  
Pemberian status klub oleh OSIS: Tidak resmi

Nama: Kasumi  
Kegiatan: Belajar bahasa inggris & menerima tantangan bermain  
Kelas: 2  
Sekolah: Sekolah Putri  
Klub: Klub bermain  
Kegiatan: Random  
Pemberian status klub oleh OSIS: Tidak resmi

BUILD SUCCESSFUL (total time: 0 seconds)