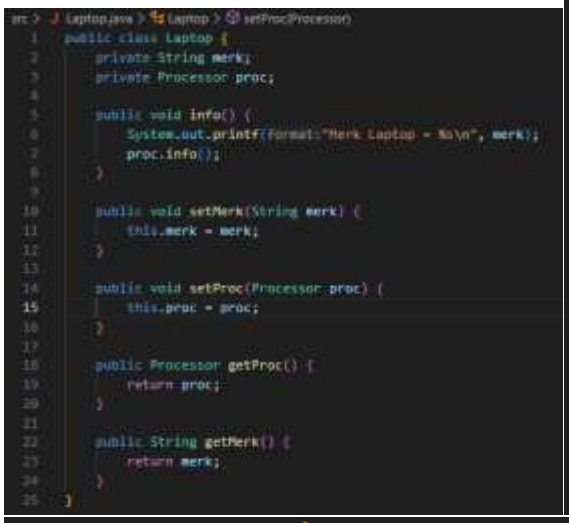
Zid'Avwa Al Bari'i

244107020083 / TI-2I / 26

Jobsheet 4

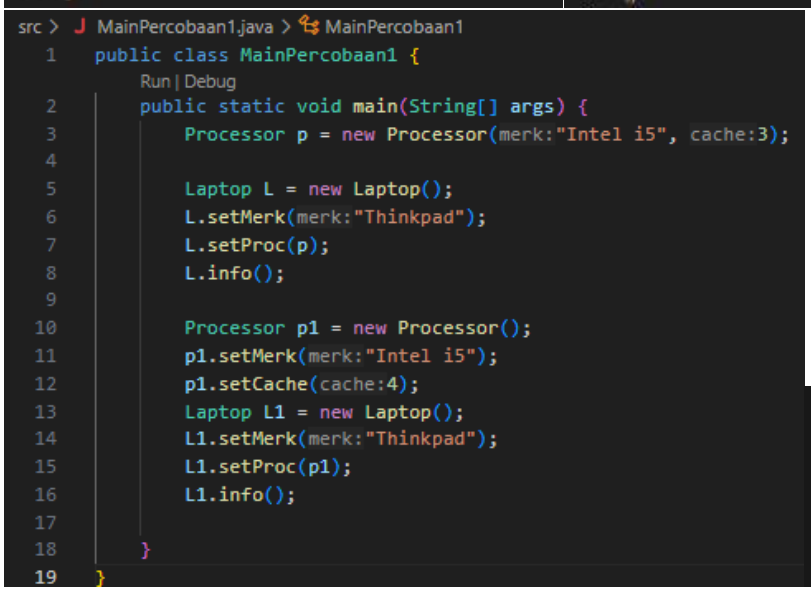
Practicum 1

Code & Output



```java
public class Laptop {
    private String merk;
    private Processor proc;

    public void info() {
        System.out.printf("Merk Laptop = %s\n", merk);
        proc.info();
    }

    public void setMerk(String merk) {
        this.merk = merk;
    }

    public void setProc(Processor proc) {
        this.proc = proc;
    }

    public Processor getProc() {
        return proc;
    }

    public String getMerk() {
        return merk;
    }
}
```

```java
public class Processor {
    private String merk;
    private double cache;

    public Processor(String merk, double cache) {
        this.merk = merk;
        this.cache = cache;
    }

    public Processor() {
    }

    public String getMerk() {
        return merk;
    }

    public double getCache() {
        return cache;
    }

    public void setMerk(String merk) {
        this.merk = merk;
    }

    public void setCache(double cache) {
        this.cache = cache;
    }

    public void info() {
        System.out.printf("Processor Merk = %s\n", merk);
        System.out.printf("Cache Memory = %.2f\n", cache);
    }
}
```



```java
public class MainPercobaan1 {
    Run | Debug
    public static void main(String[] args) {
        Processor p = new Processor(merk:"Intel i5", cache:3);

        Laptop L = new Laptop();
        L.setMerk(merk:"Thinkpad");
        L.setProc(p);
        L.info();

        Processor p1 = new Processor();
        p1.setMerk(merk:"Intel i5");
        p1.setCache(cache:4);
        Laptop L1 = new Laptop();
        L1.setMerk(merk:"Thinkpad");
        L1.setProc(p1);
        L1.info();

    }
}
```

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

Questions

1. In the Processor and Laptop classes, there are setter and getter methods for each attribute. What is the purpose of the setter and getter methods?
2. In the Processor and Laptop classes, each has a default constructor and a parameterized constructor. What is the difference in usage between these two types of constructors?
3. In the Laptop class, between the two attributes (merk and proc), which one is of type object?
4. In the Laptop class, on which line does it show that the Laptop class has a relationship with the Processor class?
5. In the Laptop class, what is the purpose of the syntax proc.info()?

6. In the MainPercobaan1 class, there is a line of code: Laptop l = new Laptop("Thinkpad", p); What is p? And what happens if that line is changed to: Laptop l = new Laptop("Thinkpad", new Processor("Intel i5", 3)); What will be the result when the program is run, is there any change?

Answers

1. Setter and getter methods are used to set and get the values of private attributes from outside the class, ensuring encapsulation.
2. The default constructor initializes objects without parameters, while the parameterized constructor allows setting attribute values during object creation.
3. In class Laptop, the attribute proc is of type object (specifically, a Processor object).
4. The line proc.info(); in Laptop shows the relationship, as it calls a method from the Processor class.
5. The syntax proc.info() is used to display information about the Processor object associated with the Laptop.
6. In Laptop l = new Laptop("Thinkpad", p);, p is a Processor object. If changed to Laptop l = new Laptop("Thinkpad", new Processor("Intel i5", 3));, it directly creates a new Processor object. The program output remains the same, but the Processor object is created inline.

Practicum 2

Code & Output

src > Mobil.java > Mobil

```java
public class Mobil {
    private String merk;
    private int biaya;

    public Mobil(){}

    public String getMerk() {
        return merk;
    }

    public int getBiaya() {
        return biaya;
    }

    public void setMerk(String merk) {
        this.merk = merk;
    }

    public void setBiaya(int biaya) {
        this.biaya = biaya;
    }

    public int hitungBiayaMobil(int hari){
        return biaya * hari;
    }
}
```

src > Sopir.java > Sopir

```java
public class Sopir {
    private String nama;
    private int biaya;

    public Sopir(){}

    public String getNama() {
        return nama;
    }

    public int getBiaya() {
        return biaya;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public void setBiaya(int biaya) {
        this.biaya = biaya;
    }

    public int hitungBiayaSopir(int hari){
        return biaya * hari;
    }
}
```

src > Pelanggan.java > Pelanggan > hitungBiayaTotal()

```java
public class Pelanggan {
    private String nama;
    private Mobil mobil;
    private Sopir sopir;
    private int hari;

    public Pelanggan() {}

    public String getNama() {
        return nama;
    }

    public Mobil getMobil() {
        return mobil;
    }

    public Sopir getSopir() {
        return sopir;
    }

    public int getHari() {
        return hari;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public void setMobil(Mobil mobil) {
        this.mobil = mobil;
    }

    public void setSopir(Sopir sopir) {
        this.sopir = sopir;
    }

    public void setHari(int hari) {
        this.hari = hari;
    }

    public int hitungBiayaTotal() {
        return mobil.hitungBiayaMobil(hari) +
               sopir.hitungBiayaSopir(hari);
    }
}
```

src > MainPercobaan2.java > ...

```java
public class MainPercobaan2 {
    Run | Debug
    public static void main(String[] args) {
        Mobil m = new Mobil();
        m.setMerk(merk:"Avanza");
        m.setBiaya(biaya:350000);
        Sopir s = new Sopir();
        s.setNama(nama:"John Doe");
        s.setBiaya(biaya:200000);
        Pelanggan p = new Pelanggan();
        p.setNama(nama:"Jane Doe");
        p.setMobil(m);
        p.setSopir(s);
        p.setHari(hari:2);
        System.out.println("Biaya Total = " + p.hitungBiayaTotal());
    }
}
```

```
Biaya Total = 1100000
```

Questions

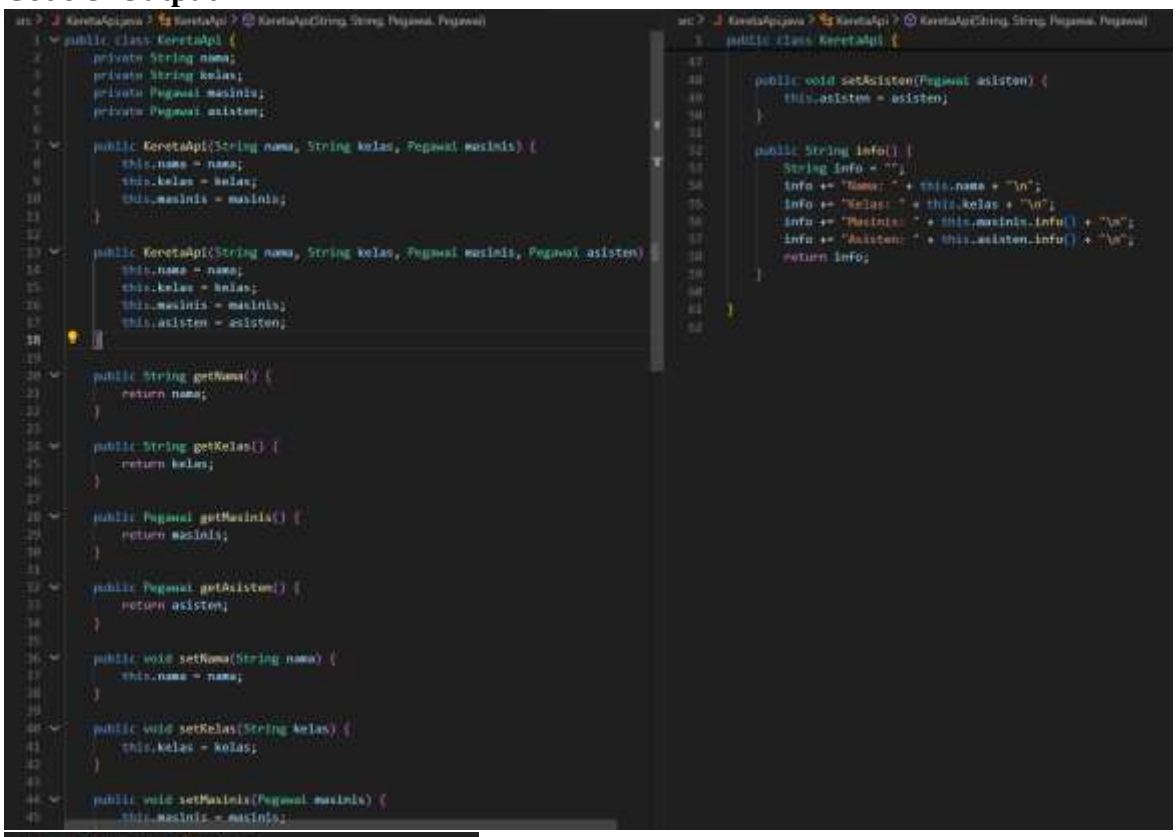
1. Observe the Pelanggan class. On which line(s) does it show that the Pelanggan class has a relationship with the Mobil and Sopir classes?
2. Observe the hitungBiayaSopir method in the Sopir class, and the hitungBiayaMobil method in the Mobil class. Why do you think these methods have the hari (days) argument?
3. Observe the code in the Pelanggan class. What is the purpose of the statements mobil.hitungBiayaMobil(hari) and sopir.hitungBiayaSopir(hari)?
4. Observe the MainPercobaan2 class. What is the purpose of the statements p.setMobil(m) and p.setSopir(s)?
5. Observe the MainPercobaan2 class. What is the purpose of the process p.hitungBiayaTotal()?
6. Observe the MainPercobaan2 class. Try adding the following line at the end of the main method and see what changes when you run it: System.out.println(p.getMobil().getMerk()); So, what does the statement p.getMobil().getMerk() in the main method do?

Answers

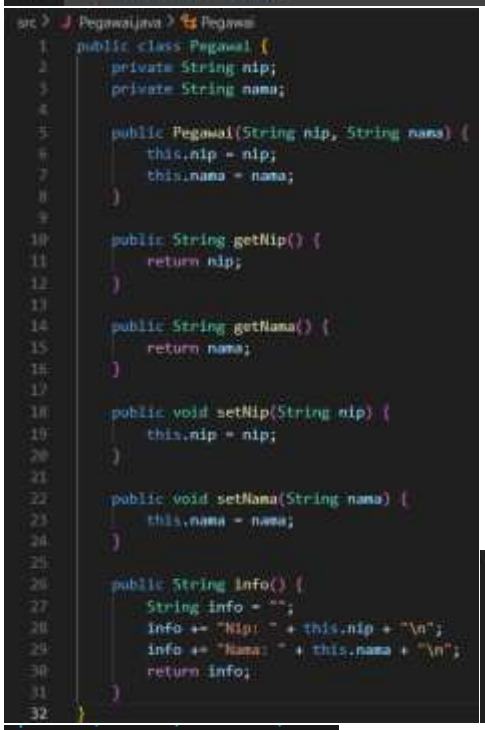
1. In the Pelanggan class, the relationship with the Mobil and Sopir classes is shown by the attributes private Mobil mobil; and private Sopir sopir;, as well as the setMobil(Mobil) and setSopir(Sopir) methods.
2. The hitungBiayaSopir and hitungBiayaMobil methods have the hari (days) argument because the cost is calculated based on the number of days used.
3. In the Pelanggan class, the statements mobil.hitungBiayaMobil(hari) and sopir.hitungBiayaSopir(hari) are used to calculate the total rental cost for the car and driver according to the number of days.
4. In the MainPercobaan2 class, the statements p.setMobil(m) and p.setSopir(s) are used to associate the Pelanggan object with the Mobil and Sopir objects.
5. In the MainPercobaan2 class, the process p.hitungBiayaTotal() is used to calculate the total rental cost for the car and driver based on the number of days.
6. If you add System.out.println(p.getMobil().getMerk()); at the end of the main method, it will display the car's brand ("Avanza") in the output.

Practicum 3

Code & Output



```java
public class KeretaApi {
    private String nama;
    private String kelas;
    private Pegawai masinis;
    private Pegawai asisten;

    public KeretaApi(String nama, String kelas, Pegawai masinis) {
        this.nama = nama;
        this.kelas = kelas;
        this.masinis = masinis;
    }

    public KeretaApi(String nama, String kelas, Pegawai masinis, Pegawai asisten) {
        this.nama = nama;
        this.kelas = kelas;
        this.masinis = masinis;
        this.asisten = asisten;
    }

    public String getNama() {
        return nama;
    }

    public String getKelas() {
        return kelas;
    }

    public Pegawai getMasinis() {
        return masinis;
    }

    public Pegawai getAsisten() {
        return asisten;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public void setKelas(String kelas) {
        this.kelas = kelas;
    }

    public void setMasinis(Pegawai masinis) {
        this.masinis = masinis;
    }

    public void setAsisten(Pegawai asisten) {
        this.asisten = asisten;
    }

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



```java
public class Pegawai {
    private String nip;
    private String nama;

    public Pegawai(String nip, String nama) {
        this.nip = nip;
        this.nama = nama;
    }

    public String getNip() {
        return nip;
    }

    public String getNama() {
        return nama;
    }

    public void setNip(String nip) {
        this.nip = nip;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String info() {
        String info = "";
        info += "Nip: " + this.nip + "\n";
        info += "Nama: " + this.nama + "\n";
        return info;
    }
}
```

```java
public class MainPercobaan1 {
    public static void main(String[] args) {
        Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");
        Pegawai asisten = new Pegawai("4567", "Patrick Star");
        KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis, asisten);
        System.out.println(keretaApi.info());
    }
}
```

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

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

Questions

1. In the info method of the KeretaApi class, what are the purposes of the lines this.masinis.info() and this.asisten.info()?
2. Create a new main program named MainPertanyaan in the same package. Add the following code to the main() method!
3. What is the output of the main program? Why does this happen?
4. Fix the KeretaApi class so that the program can run!

Answers

1. The lines this.masinis.info() and this.asisten.info() in the info() method of KeretaApi are used to display the information of the masinis (engine driver) and asisten (assistant) by calling their info() methods from the Pegawai class.
2. The output of the main program will cause a NullPointerException because the asisten field is not initialized (null) when using the constructor KeretaApi(String nama, String kelas, Pegawai masinis), but the info() method still tries to call this.asisten.info().

```
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "Pegawai.info()" because "this.asisten" is null
        at KeretaApi.info(KeretaApi.java:57)
        at MainPertanyaan.main(MainPertanyaan.java:5)
```

3. This happens because the asisten object is null, so calling a method on it causes an error.
4. To fix it, modify the info() method in KeretaApi to check if asisten is not null before calling its info() method. For example:
   info += "Asisten: " + (this.asisten != null ? this.asisten.info() : "Tidak ada") + "\n";

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

Asisten: Tidak ada
```

Practicum 4

Code & Output

```java
public class Gerbong {
    private String kode;
    private Kursi[] arrayKursi;

    public Kursi[] getArrayKursi() {
        return arrayKursi;
    }

    public String getKode() {
        return kode;
    }

    public void setKode(String kode) {
        this.kode = kode;
    }

    private void initKursi() {
        for (int i = 0; i < arrayKursi.length; i++) {
            this.arrayKursi[i] = new Kursi(String.valueOf(i + 1));
        }

    }

    public Gerbong(String kode, int jumlah) {
        this.kode = kode;
        this.arrayKursi = new Kursi[jumlah];
        this.initKursi();
    }

    public String info() {
        String info = "";
        info += "Kode: " + kode + "\n";
        for (Kursi kursi : arrayKursi) {
            info += kursi.info();
        }
        return info;
    }

    public void setPenumpang(Penumpang penumpang, int nomor) {
        this.arrayKursi[nomor - 1].setPenumpang(penumpang);
    }
}
```

```java
public class Kursi {
    private String nomor;
    private Penumpang penumpang;

    public Kursi(String nomor) {
        this.nomor = nomor;
        this.penumpang = null;
    }

    public void setPenumpang(Penumpang penumpang) {
        this.penumpang = penumpang;
    }

    public void setNomor(String nomor) {
        this.nomor = nomor;
    }

    public Penumpang getPenumpang() {
        return penumpang;
    }

    public String getNomor() {
        return nomor;
    }

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

```java
public class Penumpang {
    private String ktp;
    private String nama;

    public Penumpang(String ktp, String nama) {
        this.ktp = ktp;
        this.nama = nama;
    }

    public void setKtp(String ktp) {
        this.ktp = ktp;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getKtp() {
        return ktp;
    }

    public String getNama() {
        return nama;
    }

    public String info() {
        String info = "";
        info += "Ktp: " + ktp + "\n";
        info += "Nama: " + nama + "\n";
        return info;
    }
}
```

```java
public class MainPercobaan4 {
    Run | Debug
    public static void main(String[] args) {
        Penumpang p = new Penumpang(ktp:"12345", nama:"Mr. Krab");
        Gerbong gerbong = new Gerbong(kode:"A", jumlah:10);
        gerbong.setPenumpang(p, nomor:1);
        System.out.println(gerbong.info());
    }
}
```

```
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
```

Questions

1. There are 10 seats in Gerbong A, as specified by new Gerbong("A", 10).
2. The code checks if the seat has a passenger; if so, it adds the passenger's info to the output.
3. The seat number is reduced by 1 because array indices start from 0, but seat numbers start from 1.
4. The new passenger object budi will occupy seat 1 in the gerbong, replacing any previous passenger in that seat.
5. Modify the program so that a seat cannot be occupied if it already has a passenger.

Answers

1. In the main program in the MainPercobaan4 class, how many seats are there in Gerbong A?
2. Observe the code snippet in the info() method in the Kursi class. What does that code mean?
3. Why is the value of nomor (number) reduced by 1 in the setPenumpang() method in the Gerbong class?
4. Instantiate a new object budi of type Penumpang, then insert this new object into the gerbong with gerbong.setPenumpang(budi, 1). What happens?
5. Modify the program so that it is not allowed to occupy a seat that already has another passenger!

```
39      // public void setPenumpang(Penumpang penumpang, int nomor) {
40      // this.arrayKursi[nomor - 1].setPenumpang(penumpang);
41      // }
42
43      public void setPenumpang(Penumpang penumpang, int nomor) {
44          if (this.arrayKursi[nomor - 1].getPenumpang() == null) {
45              this.arrayKursi[nomor - 1].setPenumpang(penumpang);
46          } else {
47              System.out.println(x:"Seat already occupied!");
48          }
49      }
```

```
Seat already occupied!
Kode: A
Nomor: 1
Penumpang: Ktp: 12345
Nama: Mr. Krab
```