

OOP

Jobsheet 4



Name: Steven Christian Susanto

NIM/Number: 2241720003/24

Class: 2I

Major: Information Technology

Study Program: D-IV Informatics Engineering

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 ?
 - a. Setter digunakan untuk menentukan nilai dari merk(laptop), processor, merk(processor), dan cache.
2. Di dalam class Processor dan class Laptop, masing-masing terdapat konstruktor default dan konstruktor berparameter. Bagaimanakah beda penggunaan dari kedua jenis konstruktor tersebut ?
 - a. Konstruktor default tidak memiliki parameter sedangkan konstruktor berparameter memiliki parameter yang berfungsi untuk mendapatkan nilai saat method dipanggil.
3. Perhatikan class Laptop, di antara 2 atribut yang dimiliki (merk dan proc), atribut manakah yang bertipe object ?
 - a. Proc yang berasal dari class Processor.
4. Perhatikan class Laptop, pada baris manakah yang menunjukkan bahwa class Laptop memiliki relasi dengan class Processor ?
 - a.

```
private Processor proc;
```
5. Perhatikan pada class Laptop , Apakah guna dari sintaks proc.info() ?
 - a. Untuk meng-output-kan detail dari processor (Merk dan Cache).
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: Page 4 of 10 Laptop l = new Laptop("Thinkpad", new

Processor("Intel i5", 3)); Bagaimanakah hasil program saat dijalankan, apakah ada perubahan ?

- a. p tersebut merupakan object processor, disini tidak ada perubahan karena nilai object yang dideklarasikan di dalam parameter konstruktor laptop sama dengan nilai p.

Percobaan 2

1. Perhatikan class Pelanggan. Pada baris program manakah yang menunjukkan bahwa class Pelanggan memiliki relasi dengan class Mobil dan class Sopir ?
 - a. `private Mobil mobil;`
 - b. `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 ?
 - a. hari digunakan untuk menghitung biaya mobil (`biaya * hari`).
3. Perhatikan kode dari class Pelanggan. Untuk apakah perintah `mobil.hitungBiayaMobil(hari)` dan `sopir.hitungBiayaSopir(hari)` ?
 - a. Untuk mendapatkan biaya mobil dan biaya sopir yang kemudian akan digunakan untuk proses output.
4. Perhatikan class MainPercobaan2. Untuk apakah sintaks `p.setMobil(m)` dan `p.setSopir(s)` ?
 - a. Untuk menentukan nilai mobil dan sopir.
5. Perhatikan class MainPercobaan2. Untuk apakah proses `p.hitungBiayaTotal()` tersebut ?

- a. Untuk menghitung biaya total dari biaya mobil dan 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());
```

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

- a. Untuk mendapatkan nilai dari Mobil dan Merk.

Percobaan 3

1. Di dalam method `info()` pada class `KeretaApi`, baris `this.masinis.info()` dan `this.asisten.info()` digunakan untuk apa ?
 - a. Untuk mendapatkan info seputar 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());
```

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

a. Output

```
"C:\Program Files\Java\jdk1.8.0_121\bin\java.exe" ...  
Exception in thread "main" java.lang.NullPointerException Create breakpoint  
    at Experiment3.KeretaApi.info(KeretaApi.java:59)  
    at Experiment3.MainPertanyaan.main(MainPertanyaan.java:7)  
  
Process finished with exit code 1
```

b. Hal ini terjadi karena tidak ada object asisten, sedangkan di dalam output terdapat asisten yang harus ditampilkan.

4. Perbaiki class KeretaApi sehingga program dapat berjalan !

a. Output

```
"C:\Program Files\Java\jdk1.8.0_121\bin\java.exe" ..  
Nama: Gaya Baru  
Kelas: Bisnis  
Masinis: Nip: 1234  
Nama: Spongebob Squarepants  
  
Process finished with exit code 0
```

b. Code

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

```
        return info;
    }
}
```

Percobaan 4

1. Pada main program dalam class MainPercobaan4, berapakah jumlah kursi dalam Gerbong A ?
 - 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";
}
```

...

- a. Jika nilai penumpang tidak null maka tambahkan Penumpun: (nama_penumpang) ke dalam String.
3. Mengapa pada method setPenumpang() dalam class Gerbong, nilai nomor dikurangi dengan angka 1 ?
 - a. Karena perhitungan index dimulai dari 0 sedangkan perhitungan kursi dimulai dari 1.
 4. Instansiasi objek baru budi dengan tipe Penumpang, kemudian masukkan objek baru tersebut pada gerbong dengan gerbong.setPenumpang(budi, 1). Apakah yang terjadi ?

```
"C:\Program Files\Java\jdk1.8.0_121\bin\java.exe" ...
```

```
Kode: A
```

```
Nomor: 1
```

```
Penumpang: Ktp: 1238
```

- a. Nama: Budi

- b. Penumpang sebelumnya(Mr. Krab) tergantikan oleh Budi.
5. Modifikasi program sehingga tidak diperkenankan untuk menduduki kursi yang sudah ada penumpang lain !

a. Code

```
public void setPenumpang(Penumpang penumpang, int nomor){  
    if(this.arrayKursi[nomor - 1].getPenumpang() == null){  
        this.arrayKursi[nomor - 1].setPenumpang(penumpang);  
    } else {  
        System.out.println("Kursi ini telah terisi");  
    }  
}
```

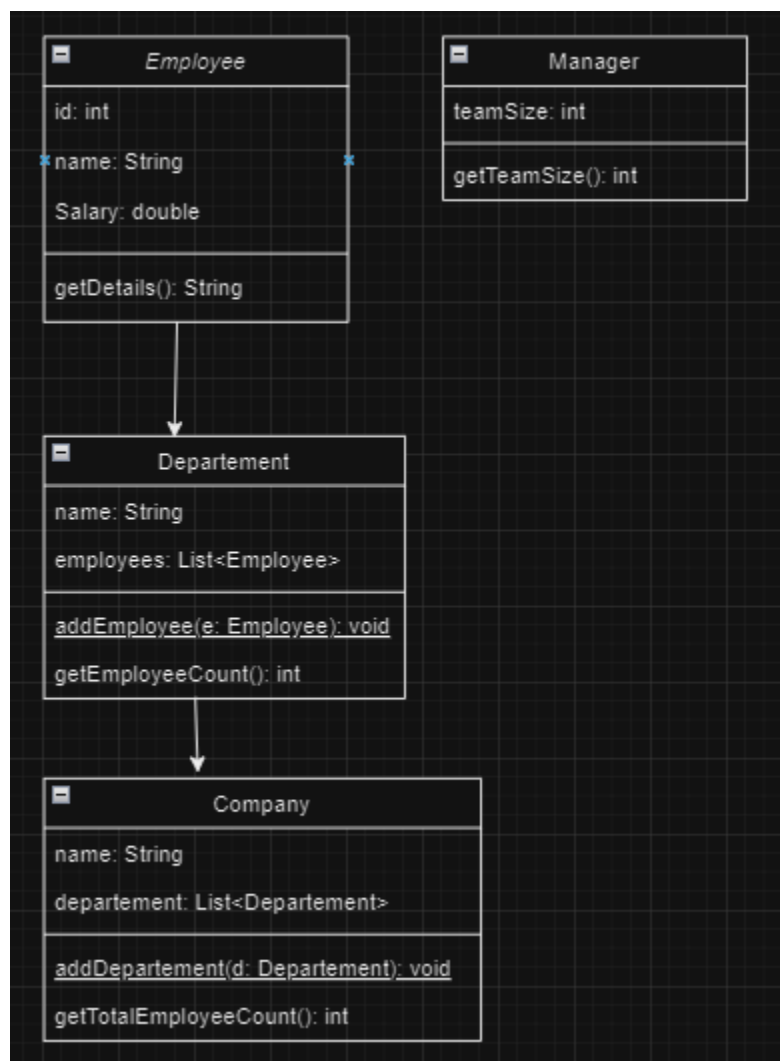
b. Output

```
"C:\Program Files\Java\jdk1.8.0_121\bin\java.exe" ...  
Kursi ini telah terisi  
Kode: A  
Nomor: 1  
Penumpang: Ktp: 12345  
Nama: Mr. Krab  
  
Nomor: 2  
Nomor: 3  
Nomor: 4
```

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)

Diagram:



Code:

```
import java.util.ArrayList;

import java.util.List;

class Employee {

    private int id;

    private String name;

    private double salary;


    public Employee(int id, String name, double salary) {

        this.id = id;

        this.name = name;

        this.salary = salary;

    }


    public String getDetails() {

        return "Employee #" + id + ": " + name + ", Salary: $" + salary;

    }

}


class Manager extends Employee {

    private int teamSize;


    public Manager(int id, String name, double salary, int teamSize) {

        super(id, name, salary);

        this.teamSize = teamSize;

    }

}
```

```
    public int getTeamSize() {  
        return teamSize;  
    }  
}
```

```
class Department {  
    private String name;  
    private List<Employee> employees;  
  
    public Department(String name) {  
        this.name = name;  
        this.employees = new ArrayList<>();  
    }  
  
    public void addEmployee(Employee employee) {  
        employees.add(employee);  
    }  
  
    public int getEmployeeCount() {  
        return employees.size();  
    }  
}
```

```
class Company {  
    private String name;
```

```

private List<Department> departments;

public Company(String name) {

    this.name = name;

    this.departments = new ArrayList<>();

}

public void addDepartment(Department department) {

    departments.add(department);

}

public int getTotalEmployeeCount() {

    int total = 0;

    for (Department department : departments) {

        total += department.getEmployeeCount();

    }

    return total;

}

}

public class Main {

    public static void main(String[] args) {

        Employee employee1 = new Employee(1, "John", 50000);

        Employee employee2 = new Employee(2, "Jane", 45000);

        Manager manager1 = new Manager(3, "Mike", 60000, 5);

```

```
Manager manager2 = new Manager(4, "Sara", 65000, 8);

Department salesDept = new Department("Sales");
Department hrDept = new Department("Human Resources");

salesDept.addEmployee(employee1);
salesDept.addEmployee(employee2);
hrDept.addEmployee(manager1);
hrDept.addEmployee(manager2);

Company myCompany = new Company("ABC Corp");
myCompany.addDepartment(salesDept);
myCompany.addDepartment(hrDept);

System.out.println("Total employees in " + myCompany.getName() + ": " +
myCompany.getTotalEmployeeCount());

}

}
```