



**LATIHAN EVALUASI AKHIR SEMESTER**  
**SEMESTER GENAP 2024/2025**  
**DEPARTEMEN MATEMATIKA - FSAD ITS**  
**PROGRAM SARJANA**



**Aturan Pengerjaan:**

- Dilarang bekerja sama dalam bentuk apa pun. Segala jenis pelanggaran (mencontek, kerjasama, dsb) yang dilakukan saat EAS akan dikenakan sanksi pembatalan mata kuliah pada semester yang sedang berjalan.
- Tuliskan Pakta Integritas di awal lembar jawaban Anda, sebagai berikut: “Dengan ini saya menyatakan bahwa saya mengerjakan sendiri tanpa bantuan dan membantu orang lain dalam menyelesaikan soal-soal EAS Alpro 2” dan ditandatangani.

Kerjakan dari yang termudah dulu yak !

1. (Skor: 25) Implementasikan konsep *class* dan *object*, *constructor*, *encapsulation*, dan *inheritance* berikut. Buat dua **class** dalam Java:

- **class** `Person` dengan atribut `name` (`String`) dan `age` (`int`). Buat *constructor* penuh untuk menginisialisasi keduanya. Semua atribut bersifat *private*, dan sediakan *getter* serta *setter* untuk masing-masing.
- **class** `Student` yang *extends* `Person`. Tambahkan atribut `studentId` (`String`) dan `gpa` (`double`). Buat *constructor* yang memanggil *super(...)* untuk `name` dan `age`, kemudian menginisialisasi `studentId` dan `gpa`. Semua atribut `Student` juga *private*, dengan *getter* dan *setter*.

Setelah itu, dalam metode `main`, buat satu objek `Student` dengan `name` = "Budi", `age` = 20, `studentId` = "S12345", `gpa` = 3.75. Tampilkan semua informasi `Student` tersebut (gunakan `System.out.println(...)`).

```
1 // Lengkapi kode di bawah ini:
2
3 public class Person {
4     private String name;
5     private int age;
6
7     // Constructor penuh
8     public Person( ... ) {
9         ...
10    }
11
12    // Getter dan Setter
13    public String getName() { ... }
14    public void setName(String name) { ... }
15    public int getAge() { ... }
16    public void setAge(int age) { ... }
17 }
18
19 public class Student extends Person {
20     private String studentId;
21     private double gpa;
22
23     // Constructor
24     public Student( ... ) {
25         // Panggil super(...) di sini
26         ...
27     }
28
29     // Getter dan Setter untuk studentId dan gpa
30     public String getStudentId() { ... }
31     public void setStudentId(String studentId) { ... }
32     public double getGpa() { ... }
33     public void setGpa(double gpa) { ... }
34 }
35
36 public class TestStudent {
37     public static void main(String[] args) {
38         // Buat objek Student dengan nama "Budi", age 20, studentId "S12345", gpa 3.75
```



**LATIHAN EVALUASI AKHIR SEMESTER**  
**SEMESTER GENAP 2024/2025**  
**DEPARTEMEN MATEMATIKA - FSAD ITS**  
**PROGRAM SARJANA**



```
39     ...
40     // Tampilkan semua informasi Student
41     ...
42 }
43 }
```

---

2. (Skor: 25) Perhatikan potongan kode di bawah, yang menggabungkan *interface* dan *abstract class*. Terdapat beberapa kesalahan (*compile error*) terkait *abstract method*, *implementation*, dan *return type*. Tugas Anda adalah menemukan dan memperbaiki kesalahan-kesalahan tersebut sehingga kode dapat dikompilasi dan dijalankan dengan benar.

---

```
1 // Interface untuk entitas yang dapat menghitung volume
2 interface Volume {
3     double calculateVolume(); // abstrak
4 }
5
6 // Abstract class untuk bangun ruang
7 abstract class Shape {
8     protected String color;
9
10    public Shape(String color) {
11        this.color = color;
12    }
13
14    // Abstract method untuk menghitung luas permukaan
15    public abstract double calculateSurfaceArea();
16
17    // Method non-abstract mencetak informasi warna
18    public void printColor() {
19        System.out.println("Color: " + color);
20    }
21 }
22
23 // Class Box yang harus meng-implement interface Volume dan menurunkan Shape
24 public class Box extends Shape implements Volume {
25     private double length;
26     private double width;
27     private double height;
28
29     // Constructor
30     public Box(String color, double length, double width, double height) {
31         // Panggil super constructor
32         ...
33         this.length = length;
34         this.width = width;
35         this.height = height;
36     }
37
38     // Lengkapi implementasi abstract method calculateSurfaceArea()
39     public ... calculateSurfaceArea() {
40         // Rumus: 2*(lw + lh + wh)
41         return ...;
42     }
43
44     // Lengkapi implementasi interface calculateVolume()
45     public ... calculateVolume() {
46         // Rumus: length * width * height
47         return ...;
48     }
49
50     // Tambahkan main untuk menguji:
51     // Buat objek Box dengan color "Red", length 2.0, width 3.0, height 4.0,
52     // Lalu panggil calculateSurfaceArea() dan calculateVolume(), serta printColor()
53 }
```

---

3. (Skor: 25) Berdasarkan spesifikasi berikut, buat class *Car*, kemudian buat satu instance dari *Car* di dalam *main*, dan tampilkan hasilnya:



**LATIHAN EVALUASI AKHIR SEMESTER**  
**SEMESTER GENAP 2024/2025**  
**DEPARTEMEN MATEMATIKA - FSAD ITS**  
**PROGRAM SARJANA**



- Car memiliki atribut `brand` (String), `model` (String), `year` (int), dan `mileage` (double). Semua atribut *private*.
- Buat *constructor* dengan parameter `brand`, `model`, `year`; `mileage` diinisialisasi default 0.0.
- Buat *getter* untuk semua atribut, dan *setter* hanya untuk `mileage`.
- Buat method `drive(double km)` yang menambahkan nilai `mileage` dengan `km`.
- Buat method `toString()` yang mengembalikan String dengan format: "{brand} {model}, Year: {year}, Mileage: {mileage} km" (misal: "Toyota Corolla, Year: 2020, Mileage: 15000.5 km").

*Contoh:*

Buat sebuah objek `Car` dengan `brand` = "Toyota", `model` = "Corolla", `year` = 2020. Kemudian panggil `drive(15000.5)` dan cetak `toString()`, hasilnya harus menampilkan: Toyota Corolla, Year: 2020, Mileage: 15000.5 km

---

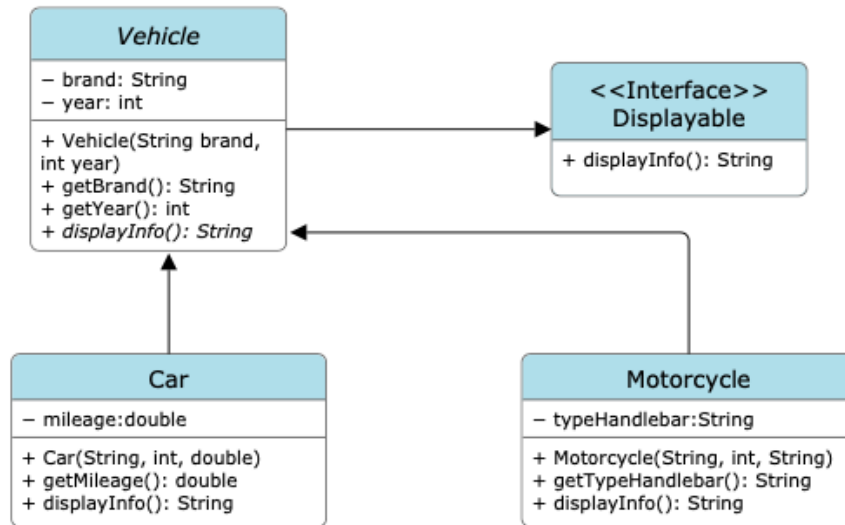
```
1 // Lengkapi kode di bawah ini:
2
3 public class Car {
4     private String brand;
5     private String model;
6     private int year;
7     private double mileage;
8
9     // Constructor
10    public Car( ... ) {
11        ...
12    }
13
14    // Getter untuk semua atribut
15    public String getBrand() { ... }
16    public String getModel() { ... }
17    public int getYear() { ... }
18    public double getMileage() { ... }
19
20    // Setter hanya untuk mileage
21    public void setMileage( ... ) { ... }
22
23    // Method drive untuk menambahkan mileage
24    public void drive(double km) {
25        ...
26    }
27
28    // Override toString()
29    @Override
30    public String toString() {
31        ...
32    }
33
34    // Main untuk menguji
35    public static void main(String[] args) {
36        // Buat objek Car brand "Toyota", model "Corolla", year 2020
37        Car myCar = new Car(...);
38        // Panggil drive(15000.5)
39        myCar.drive(15000.5);
40        // Cetak hasil toString()
41        System.out.println(myCar.toString());
42    }
43 }
```

---

4. (Skor: 25) Analisis diagram UML berikut yang mencakup konsep OOP mulai dari *inheritance*, *abstract class*, hingga *interface*. Jawablah pertanyaan di bawahnya.



LATIHAN EVALUASI AKHIR SEMESTER  
SEMESTER GENAP 2024/2025  
DEPARTEMEN MATEMATIKA - FSAD ITS  
PROGRAM SARJANA



- Sebutkan tipe relasi antara **Vehicle**, **Car**, dan **Motorcycle**. Jelaskan singkat.
- Mengapa **Vehicle** dideklarasikan sebagai *abstract class* dan bukan *interface*?
- Tulis kode Java singkat untuk **class Car** yang mengimplementasikan semua metode yang diperlukan sesuai UML di atas. Sertakan *keyword* yang tepat (*extends*, *implements*, *@Override*, dll.).
- Bagaimana hubungan **Vehicle** dengan **Displayable**? Jelaskan bagaimana kelas-kelas di atas memenuhi kontrak **Displayable**.
- Jika ingin menambahkan **class Truck** yang juga mewarisi **Vehicle** dan menambahkan atribut `loadCapacity` (double), bagaimana Anda menuliskan *constructor* dan *method displayInfo()* untuk **Truck**? Berikan contoh kode singkat.