

## Pengantar Konsep Pemrograman Berorientasi Objek

### 1. Kompetensi

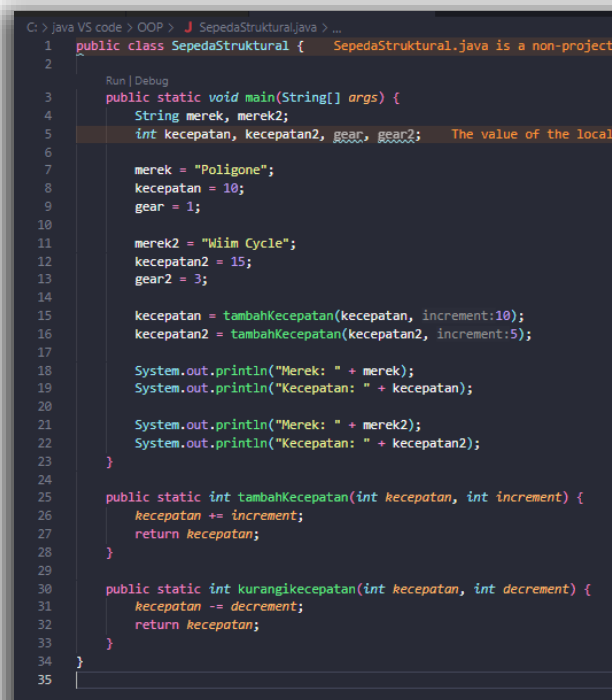
Setelah menempuh materi percobaan ini, mahasiswa mampu mengenal memahami kembali konsep pemrograman dengan **paradigma struktural (structured programming)**

### 2. Pendahuluan

Perbedaan mendasar antara pemrograman prosedural dan pemrograman berorientasi objek (PBO) atau *Object Oriented Programming* (OOP) adalah:

- Pemrograman terstruktur: program dipecah kedalam sub-program berupa **fungsi**. Karakteristik dari objek (apa yang dimiliki **dan** apa yang dilakukan) direpresentasikan dalam variable dan fungsi yang **berdiri sendiri** (tidak terikat)
- PBO: program dipecah kedalam **objek**, dimana objek tersebut membungkus **atribut** dan **method**.

Berikut merupakan contoh dari pemrograman struktural:



```
C:\> java VS code > OOP > J SepedaStruktural.java > ...
1 public class SepedaStruktural { SepedaStruktural.java is a non-project
2
3 Run | Debug
4 public static void main(String[] args) {
5     String merek, merek2;
6     int kecepatan, kecepatan2, gear, gear2; The value of the local
7
8     merek = "Poligone";
9     kecepatan = 10;
10    gear = 1;
11
12    merek2 = "Wiim Cycle";
13    kecepatan2 = 15;
14    gear2 = 3;
15
16    kecepatan = tambahKecepatan(kecepatan, increment:10);
17    kecepatan2 = tambahKecepatan(kecepatan2, increment:5);
18
19    System.out.println("Merek: " + merek);
20    System.out.println("Kecepatan: " + kecepatan);
21
22    System.out.println("Merek: " + merek2);
23    System.out.println("Kecepatan: " + kecepatan2);
24
25 }
26
27 public static int tambahKecepatan(int kecepatan, int increment) {
28     kecepatan += increment;
29     return kecepatan;
30 }
31
32 public static int kurangiKecepatan(int kecepatan, int decrement) {
33     kecepatan -= decrement;
34     return kecepatan;
35 }
```

Berdasarkan contoh tersebut, dapat dilihat bahwa dalam paradigma pemrograman struktural:

1. Ciri/status/nilai dari objek berjenis sepeda (hal-hal yang dimiliki oleh sepeda) di dunia nyata direpresentasikan atau disimpan di dalam program sebagai **variabel yang berdiri sendiri atau tidak saling berkaitan**.

Sepeda yang pertama, karakteristiknya disimpan dalam variable `merek`, `kecepatan`, dan `gear`

Sepeda yang kedua, karakteristiknya disimpan dalam variable `merek2`, `kecepatan2`, dan `gear2`

Jika nantinya ada sepeda yang ketiga, kemungkinan akan disimpan dalam variable `merek3`, `kecepatan3`, dan `gear2`

Efeknya, **tidak** ada mekanisme yang **menjamin** bahwa variable `merek2`, `kecepatan2`, dan `gear2` saling terhubung

2. Prosedur/perilaku/proses dari sepeda (hal-hal yang bisa dilakukan oleh sepeda) di dunia nyata direpresentasikan sebagai fungsi yang bisa dipanggil/dieksekusi, yaitu `tambahKecepatan` dan `kurangiKecepatan()`

Tapi cara ini **tidak menjamin** bahwa kedua fungsi tersebut **hanya dapat dipanggil oleh objek berjenis sepeda**, bisa saja objek berjenis kursi bisa memanggil fungsi ini.

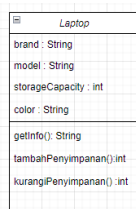
### 3. Tugas Praktikum

#### 3.1. Praktikum 1

Lakukan langkah-langkah berikut supaya tugas praktikum yang dikerjakan tersistematis:

- a. Tentukan 1 kategori objek. Anda bisa menggunakan jenis objek baru atau salah satu objek dari tugas PBO Teori.
- b. Lakukan pengamatan terhadap objek tersebut untuk menentukan
  - 4 state/ciri/status/nilai yang bisa dimiliki
  - 3 behavior/prosedur/perilaku/fungsi/proses yang dapat dilakukan objek tersebut
- c. Implementasikan 5 buah objek dari jenis tersebut ke dalam program dengan paradigma **pemrograman struktural** (seperti pada contoh sepeda di atas)
  - Deklarasikan dan inisialisasikan variable untuk setiap ciri/status/nilai dari objek sebagai variable
  - Buatlah function dari setiap prosedur/perilaku/proses yang dapat dilakukan oleh objek kemudian coba lakukan pemanggilan function tersebut

Class diagram yang dipilih:



Kode :

```

J: Laptop.java > Laptop : kurangiStorage(int, int, int)
1 public class Laptop {
2     // informasi
3     public static void main(String[] args) {
4
5         String brand1, brand2, brand3, brand4, brand5;
6         String model1, model2, model3, model4, model5;
7         int storageCapacity1, storageCapacity2, storageCapacity3, storageCapacity4, storageCapacity5;
8         String color1, color2, color3, color4, color5;
9
10
11         brand1 = "HP";
12         model1 = "Pavilion";
13         storageCapacity1 = 512;
14         color1 = "Silver";
15
16         brand2 = "Dell";
17         model2 = "Inspiron";
18         storageCapacity2 = 256;
19         color2 = "Black";
20
21         brand3 = "Asus";
22         model3 = "ZenBook";
23         storageCapacity3 = 1024;
24         color3 = "Gold";
25
26         brand4 = "Lenovo";
27         model4 = "ThinkPad";
28         storageCapacity4 = 512;
29         color4 = "Gray";
30
31         brand5 = "Apple";
32         model5 = "MacBook Pro";
33         storageCapacity5 = 512;
34         color5 = "Space Gray";
35
36         System.out.println("Informasi Laptop:");
37         printLaptopInfo(laptopNumber:1, brand1, model1, storageCapacity1, color1);
38         printLaptopInfo(laptopNumber:2, brand2, model2, storageCapacity2, color2);
39         printLaptopInfo(laptopNumber:3, brand3, model3, storageCapacity3, color3);
40         printLaptopInfo(laptopNumber:4, brand4, model4, storageCapacity4, color4);
41         printLaptopInfo(laptopNumber:5, brand5, model5, storageCapacity5, color5);
42
43         tambahStorage(laptopNumber:1, additionalCapacity:256, storageCapacity1);
44         kurangiStorage(laptopNumber:3, reducedCapacity:128, storageCapacity3);
45     }
46
47     public static void printLaptopInfo(int laptopNumber, String brand, String model, int storageCapacity, String color) {
48         System.out.println("Laptop " + laptopNumber + ":");
49         System.out.println("Merek: " + brand);
50         System.out.println("Model: " + model);
51         System.out.println("Storage Capacity: " + storageCapacity + "GB");
52         System.out.println("Color: " + color);
53         System.out.println();
54     }
55
56
57
58
59     public static void tambahStorage(int laptopNumber, int additionalCapacity, int stg) {
60         stg += additionalCapacity;
61         System.out.println("Penyimpanan pada Laptop " + laptopNumber + " ditambahkan sebanyak " + additionalCapacity + "GB.");
62         System.out.println("Kapasitas penyimpanan Laptop " + laptopNumber + " sekarang adalah " + stg + "GB.");
63     }
64
65     public static void kurangiStorage(int laptopNumber, int reducedCapacity, int stg) {
66         stg -= reducedCapacity;
67         System.out.println("Penyimpanan pada Laptop " + laptopNumber + " dikurangi sebanyak " + reducedCapacity + "GB. (karena rusak)");
68         System.out.println("Kapasitas penyimpanan Laptop " + laptopNumber + " sekarang adalah " + stg + "GB.");
69     }
70
71 }
72

```

## Output :

```

Informasi Laptop:
Laptop 1:
Merek: HP
Model: Pavilion
Storage Capacity: 512GB
Color: Silver

Laptop 2:
Merek: Dell
Model: Inspiron
Storage Capacity: 256GB
Color: Black

Laptop 3:
Merek: Asus
Model: ZenBook
Storage Capacity: 1024GB
Color: Gold

Laptop 4:
Merek: Lenovo
Model: ThinkPad
Storage Capacity: 512GB
Color: Gray

Laptop 5:
Merek: Apple
Model: MacBook Pro
Storage Capacity: 512GB
Color: Space Gray

Penyimpanan pada Laptop 1 ditambahkan sebanyak 256GB.
Kapasitas penyimpanan Laptop 1 sekarang adalah 768GB.
Penyimpanan pada Laptop 3 dikurangi sebanyak 128GB. (karena rusak)
Kapasitas penyimpanan Laptop 3 sekarang adalah 896GB.
PS C:\xampp\htdocs\dasarWeb>

```

### 3.2. Praktikum 2

Buatlah program kalkulator sederhana dengan paradigma **pemrograman struktural** yang dapat menerima input angka1, operator, dan angka2 dan menampilkan hasilnya ke console/layar

Kode :

```
C:\> java VS code > OOP > J Kalkulator.java > ...
1  import java.util.Scanner;
2
3  public class Kalkulator {
4
5      public static void main(String[] args) {
6          Scanner scanner = new Scanner(System.in);
7
8          System.out.print("Masukkan angka pertama: ");
9          double angka1 = scanner.nextDouble();
10
11         System.out.print("Masukkan operator (+, -, *, /): ");
12         char operator = scanner.next().charAt(0);
13
14         System.out.print("Masukkan angka kedua: ");
15         double angka2 = scanner.nextDouble();
16
17         double hasil = 0;
18         switch(operator) {
19             case '+':
20                 hasil = tambah(angka1, angka2);
21                 break;
22             case '-':
23                 hasil = kurang(angka1, angka2);
24                 break;
25             case '*':
26                 hasil = kali(angka1, angka2);
27                 break;
28             case '/':
29                 if (angka2 != 0) {
30                     hasil = bagi(angka1, angka2);
31                 } else {
32                     System.out.println("Pembagian dengan nol tidak dapat dilakukan.");
33                     return;
34                 }
35                 break;
36             default:
37                 System.out.println("Operator yang dimasukkan tidak valid.");
38                 return;
39         }
40
41         System.out.println("Hasil: " + hasil);
42
43         scanner.close();
44     }
45
46     public static double tambah(double angka1, double angka2) {
47         return angka1 + angka2;
48     }
49
50     public static double kurang(double angka1, double angka2) {
51         return angka1 - angka2;
52     }
53
54     public static double kali(double angka1, double angka2) {
55         return angka1 * angka2;
56     }
57
58     public static double bagi(double angka1, double angka2) {
59         return angka1 / angka2;
60     }
61 }
62
```

Output :

```
Masukkan angka pertama: 3
Masukkan operator (+, -, *, /): *
Masukkan angka kedua: 5
Hasil: 15.0
```

#### 4. Pertanyaan

Tuliskan analisa Anda apakah pemrograman dengan paradigma terstruktur sesuai digunakan untuk tugas praktikum 1 dan 2? Jelaskan!

Jawab :

Pemrograman dengan paradigma terstruktur tidak sesuai untuk tugas praktikum 1 karena keterbatasan dalam mengelola data, kode sulit dibaca, dan kesulitan dalam memperluas serta memodifikasi kode. Karena berasal dari class diagram yang dijadikan structural maka kode yang ditulis akan sangat banyak.

sedangkan Pada praktikum 2 (membuat kalkulator), penggunaan paradigma terstruktur masih memungkinkan karena sifatnya yang lebih sederhana dan tidak melibatkan manajemen data yang kompleks, sehingga kode yang dihasilkan masih dapat diorganisir dengan baik meskipun akan lebih panjang karena berasal dari class diagram yang telah disediakan.