

# Introduction to Object-Oriented Programming Concepts

Name : Fitri Cahyaniati(11)

Class : SIB1G

NIM : 2341760198

## 1. Competence

After taking this experimental material, students are able to get to know and understand the concept of programming with a **structural paradigm (structured programming)**

## 2. Introduction

The fundamental differences between procedural programming and Object Oriented Programming (OOP) are:

- Structured programming: programs are broken down into sub-programs in the form of **functions**. The characteristics of the object (what it has **and** what it does) are represented in independent (unbound) variables and functions
- PBO: The program is broken down into **objects**, where the object wraps **attributes** and **methods**.

Here is an example of structural programming:

```
bycycle.java > bycycle > main(String[])
1  public class bycycle
2  {
3      Run | Debug
4      public static void main(String[] args)
5      {
6          String merk, merk2;
7          int speed, speed2, gear, gear2;
8          merk = "Polygon";
9          speed = 10;
10         gear = 1;
11
12         merk2 = "Wiim Cycle";
13         speed2 = 15;
14         gear2 = 3;
15
16         speed = increaseSpeed(speed, increment:10);
17         speed2 = reduceSpeed(speed2, decrement:5);
18
19
20         System.out.println("merk = " + merk);
21         System.out.println("speed = " + speed);
22
23         System.out.println("merk = " + merk2);
24         System.out.println("speed = " + speed2);
25
26     }
27
28     public static int increaseSpeed(int speed, int increment)
29     {
30         speed += increment;
31         return speed;
32     }
33     public static int reduceSpeed(int speed, int decrement)
34     {
35         speed -= decrement;
36         return speed;
37     }
38 }
```

Based on these examples, it can be seen that in the structural programming paradigm:

1. The characteristics/status/values of bicycle objects (things owned by bicycles) in the real world are represented or stored in the program as **independent or unrelated variables**.

The first bike, its characteristics are stored in the variables of merk, speed, and gear

The second bicycle has its characteristics stored in the merk2, speed2, and gear2

If there is a third bike, it will likely be stored in the merk3, speed3, and gear3

In effect, **there is no mechanism that guarantees** that the brand variables2, speed2, and gear2 are interconnected

2. The procedure/behavior/process of a bicycle (things that a bicycle can do) in the real world is represented as a function that can be called/executed, i.e. increaseSpeed() and reduceSpeed()

But this method **does not guarantee** that the two functions **can only be called by a bicycle-type object**, it is possible that a seat-type object can call this function.

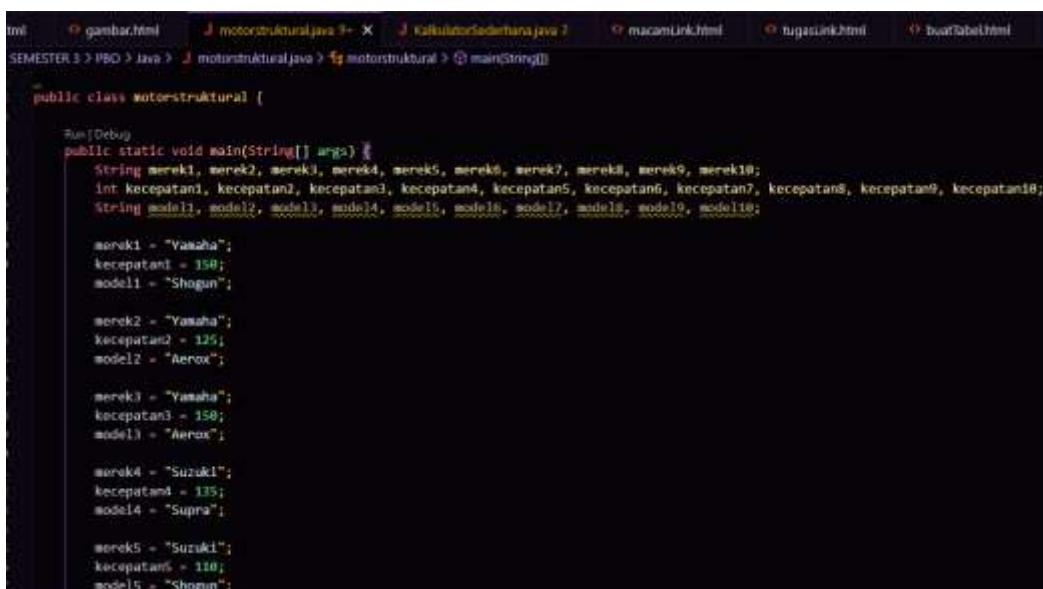
### 3. Practicum Assignment

#### 3.1. Task 1

Take the following steps so that the practicum assignments carried out are systematic:

- a. Define 1 category/class of objects. You can use a new object type or one of the objects from the PBO Theory task. For example: Bicycle
- b. Make observations of the object to determine
  - 3 variables/states/traits/states/values that can be had
  - 2 functions/behaviors/procedures/behaviors/processes that the object can perform
- c. Implement 10 objects of this type into a program with a **structural programming paradigm** (as in the bicycle example above)
  - Declare and initialize a variable for each characteristic/state/value of an object as a variable
  - Create a function of every procedure/behavior/process that can be performed by the object and then try to call the function/method
  -

**My answer :**



```
public class motorstruktural {  
    public static void main(String[] args) {  
        String merk1, merk2, merk3, merk4, merk5, merk6, merk7, merk8, merk9, merk10;  
        int kecepatan1, kecepatan2, kecepatan3, kecepatan4, kecepatan5, kecepatan6, kecepatan7, kecepatan8, kecepatan9, kecepatan10;  
        String model1, model2, model3, model4, model5, model6, model7, model8, model9, model10;  
  
        merk1 = "Yamaha";  
        kecepatan1 = 150;  
        model1 = "Shogun";  
  
        merk2 = "Yamaha";  
        kecepatan2 = 125;  
        model2 = "Aerox";  
  
        merk3 = "Yamaha";  
        kecepatan3 = 150;  
        model3 = "Aerox";  
  
        merk4 = "Suzuki";  
        kecepatan4 = 135;  
        model4 = "Supra";  
  
        merk5 = "Suzuki";  
        kecepatan5 = 110;  
        model5 = "Shogun";  
    }  
}
```

```

> SEMESTER 3 > PBO > Java > J motorstruktural.java > motorstruktural > main(String[])
2   public class motorstruktural {
4       public static void main(String[] args) {
36
37           merek8 = "Honda";
38           kecepatan8 = 125;
39           model8 = "Vario";
40
41           merek9 = "Honda";
42           kecepatan9 = 125;
43           model9 = "Filanio";
44
45           merek10 = "Honda";
46           kecepatan10 = 175;
47           model10 = "Delux";
48
49           kecepatan1 = tambahKecepatan(kecepatan1, increment:10);
50           kecepatan2 = tambahKecepatan(kecepatan2, increment:15);
51           kecepatan3 = tambahKecepatan(kecepatan3, increment:20);
52           kecepatan4 = tambahKecepatan(kecepatan4, increment:25);
53           kecepatan5 = tambahKecepatan(kecepatan5, increment:30);
54           kecepatan6 = tambahKecepatan(kecepatan6, increment:35);
55           kecepatan7 = tambahKecepatan(kecepatan7, increment:40);
56           kecepatan8 = tambahKecepatan(kecepatan8, increment:45);
57           kecepatan9 = tambahKecepatan(kecepatan9, increment:50);
58           kecepatan10 = tambahKecepatan(kecepatan10, increment:55);
59

```

```

System.out.println(x:"=====");
System.out.println(x:"Dealer motor Cahya");
System.out.println(x:"=====");

System.out.println("Merek 1: " + merek1 + ", Kecepatan: " + kecepatan1);
System.out.println("Merek 2: " + merek2 + ", Kecepatan: " + kecepatan2);
System.out.println("Merek 3: " + merek3 + ", Kecepatan: " + kecepatan3);
System.out.println("Merek 4: " + merek4 + ", Kecepatan: " + kecepatan4);
System.out.println("Merek 5: " + merek5 + ", Kecepatan: " + kecepatan5);
System.out.println("Merek 6: " + merek6 + ", Kecepatan: " + kecepatan6);
System.out.println("Merek 7: " + merek7 + ", Kecepatan: " + kecepatan7);
System.out.println("Merek 8: " + merek8 + ", Kecepatan: " + kecepatan8);
System.out.println("Merek 9: " + merek9 + ", Kecepatan: " + kecepatan9);
System.out.println("Merek 10: " + merek10 + ", Kecepatan: " + kecepatan10);
}

// Method penambahan kecepatan
public static int tambahKecepatan(int kecepatan, int increment) {
    kecepatan += increment;
    return kecepatan;
}

```

```

PS C:\Users\VIP> . & 'C:\Program Files\Java\j
ct\bin\' 'motorstruktural'

=====
Dealer motor Cahya
=====
Merek 1: Yamaha, Kecepatan: 160
Merek 2: Yamaha, Kecepatan: 148
Merek 3: Yamaha, Kecepatan: 170
Merek 4: Suzuki, Kecepatan: 160
Merek 5: Suzuki, Kecepatan: 148
Merek 6: Honda, Kecepatan: 168
Merek 7: Suzuki, Kecepatan: 140
Merek 8: Honda, Kecepatan: 160
Merek 9: Honda, Kecepatan: 168
Merek 10: Honda, Kecepatan: 195
Merek 11: Honda, Kecepatan: 175
Merek 12: Honda, Kecepatan: 175
Merek 13: Honda, Kecepatan: 230
PS C:\Users\VIP>

```

```

// Method pengurangan kecepatan
public static int kurangiKecepatan(int kecepatan, int decrement) {
    kecepatan -= decrement;
    return kecepatan;
}

```

### 3.2. Task 2

Create a simple calculator program with a **structural programming paradigm** that can accept the input of number1, operator, and number2 and display the results to the console/screen

My answer :

```
SEMESTER 3 > PBO > Java > J KalkulatorSederhana.java > KalkulatorSederhana > main(String[])
import java.util.Scanner;

public class KalkulatorSederhana {

    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print(s:"Masukkan angka pertama : ");
        double angka1 = scanner.nextDouble();

        System.out.print(s:"Masukkan operator (+, -, *, /) : ");
        char operator = scanner.next().charAt(index:0);

        System.out.print(s:"Masukkan angka kedua : ");
        double angka2 = scanner.nextDouble();

        // untuk menyimpan hasil
        double hasil = 0;

        // Melakukan operasi aritmatika
        switch (operator) {
            case '+':
                hasil = angka1 + angka2;
                break;
            case '-':
                hasil = angka1 - angka2;
```

```
            switch (operator) {
                case '+':
                    hasil = angka1 + angka2;
                    break;
                case '-':
                    hasil = angka1 - angka2;
                    break;
                case '*':
                    hasil = angka1 * angka2;
                    break;
                case '/':
                    if (angka2 != 0) {
                        hasil = angka1 / angka2;
                    } else {
                        System.out.println(x:"Error : tidak boleh pembagian nol.");
                        return;
                    }
                    break;
                default:
                    System.out.println(x:"Error : Operator tidak valid.");
                    return;
            }

            System.out.println("Hasil: " + hasil);
```

```
PS C:\Users\HP> & 'C:\Program Files\Java\jdk-21
jdt.ls-java-project\bin' 'KalkulatorSederhana'
Masukkan angka pertama : 7
Masukkan angka pertama : 7
Masukkan operator (+, -, *, /) : -
Masukkan angka kedua : 6
Hasil: 1.0
PS C:\Users\HP>
```

#### 4. Question

Write down your analysis, is programming with a structured paradigm suitable for practicum assignments 1 and 2? Explain!

**My answer :**

##### **1. Practical Assignment 1: Motor Structural**

**Analysis:**

- **Advantages:** The code is easy to understand and suitable for simple cases with a limited number of motors. This program uses a structured approach by defining variables separately and using statistical functions to increase or decrease motor speed
- **Disadvantages:** Cannot be weak, difficult to manage and maintain if the number of motors or complexity increases.

##### **2. Practical Assignment 2: Simple Calculator**

**Analysis:**

- **Advantages:** Easy to understand and focuses on basic functions.
- **Disadvantages:** Less suitable for adding new features, the structured paradigm may be less efficient and difficult to develop along with more complex needs.

**Conclusion:**

**In my opinion, the structured paradigm is suitable for Practical Assignments 1 and 2 if the cases faced are simple and the amount of data or features is limited. However, for further development or if the level of difficulty increases, a more modular/separate and scalable object-based approach may be more effective.**

## 5. Direction

1. Work on the Java program code via Netbean or Visual studio Code.
2. Go to the lecturer's desk for an oral test of the results of the practical work assignment (about 5 minutes) after finishing you can go home first or continue working on the report.
3. Report in PDF according to the practical steps and screenshots of the program code, (the original program code for each practical should be kept safe by yourself if the quiz or exam time is asked)
4. Collect the report in LMS in PDF format with a deadline of 23.59 WIB on the same day as the practical, outside of that hour it will not be corrected.

**--- Good Luck----**