

## **OOP Practicum**



Malik Abdul Azis

2241720240

1I Class

14

**STUDY PROGRAM D-IV INFORMATIC ENGINEER**  
**INFORMATION TECHNOLOGY DEPARTMENT**  
**MALANG STATE POLYTECHNIC**

Soekarno Hatta Street No.9, Jatimulyo, Lowokwaru District, Malang City, East Java

65141

## Percobaan 1: Membuat Class Diagram

### Studi Kasus 1:

Dalam suatu perusahaan salah satu data yang diolah adalah data karyawan. Setiap karyawan memiliki id, nama, jenis kelamin, jabatan, dan gaji. Setiap karyawan juga bisa menampilkan data diri pribadi dan melihat gajinya.

1. Gambarkan desain class diagram dari studi kasus 1!,

Jawab:

Karyawan
<ul style="list-style-type: none"><li>• id : int</li><li>• nama : string</li><li>• jenisKelamin : string</li><li>• jabatan : string</li><li>• gaji : double</li></ul>
<ul style="list-style-type: none"><li>❖ getDataDiri() : string</li><li>❖ getGaji() : double</li></ul>

2. Sebutkan Class apa saja yang bisa dibuat dari studi kasus 1!,

Jawab:

Karyawan: Class ini merepresentasikan entitas karyawan dengan atribut seperti id, nama, jenis kelamin, jabatan, dan gaji. Class ini juga memiliki dua method yaitu getDataDiri() untuk mendapatkan data diri karyawan dan getGaji() untuk mendapatkan gaji karyawan.

3. Sebutkan atribut beserta tipe datanya yang dapat diidentifikasi dari masing-masing class dari studi kasus 1!

Jawab:

Atribut yang berasal dari class Karyawan adalah id dengan tipe data int, nama dengan tipe data string, jenisKelamin dengan tipe data string, dan gaji dengan tipe data double.

4. Sebutkan method-method yang sudah anda buat dari masing-masing class pada studi kasus 1!

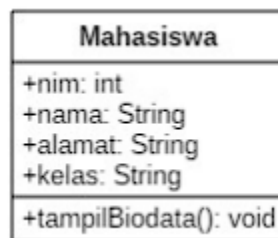
Jawab:

Method yang dibuat dari class Karyawan adalah getDataDiri() yang digunakan untuk menampilkan informasi lengkap tentang seorang karyawan seperti ID, nama, jenis kelamin, jabatan, dan gaji. Method berfungsi untuk mencetak atau menampilkan informasi karyawan ke layar atau file. Dan method getGaji() yang digunakan untuk melihat gaji karyawan tanpa menampilkan seluruh informasi pribadi karyawan.

## Percobaan 2: Membuat dan mengakses anggota suatu class

Studi Kasus 2:

Perhatikan class diagram dibawah ini. Buatlah program berdasarkan class diagram tersebut!



2.

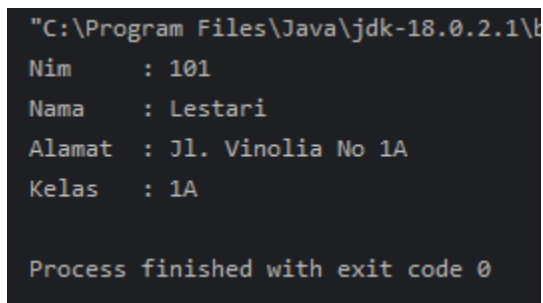
```
1 public class Mahasiswa{
2     public int nim;
3     public String nama;
4     public String alamat;
5     public String kelas;
6
7     public void tampilBiodata(){
8         System.out.println("Nim      : " +nim);
9         System.out.println("Nama      : " +nama);
10        System.out.println("Alamat   : " +alamat);
11        System.out.println("Kelas   : " +kelas);
12    }
13 }
```

4.



```
1 public class TestMahasiswa {
2     public static void main(String[] args) {
3         Mahasiswa mhs1 = new Mahasiswa();
4         mhs1.nim=101;
5         mhs1.nama="Lestari";
6         mhs1.alamat="Jl. Vinolia No 1A";
7         mhs1.kelas="1A";
8         mhs1.tampilBiodata();
9     }
10 }
11 }
```

6.



```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" -Djava.class.path=.\TestMahasiswa.class
Nim      : 101
Nama     : Lestari
Alamat  : Jl. Vinolia No 1A
Kelas   : 1A

Process finished with exit code 0
```

7. Proses pendeklarasian atribut pada program di atas terjadi di dalam class Mahasiswa.

8. Proses pendeklarasian method pada program di atas juga terjadi di dalam class Mahasiswa.

9. Hanya satu objek yang diinstansiasi pada program di atas, yaitu mhs1 dari class Mahasiswa.

10. Pada sintaks program **mhs1.nim=101**, nilai 101 dimasukkan ke dalam atribut **nim** dari objek **mhs1**. Ini adalah proses memberikan nilai pada atribut **nim** milik objek **mhs1**.

11. Pada sintaks program **mhs1.tampilBiodata()**, method **tampilBiodata()** dari objek **mhs1** dipanggil. Ini akan menjalankan kode di dalam method tersebut, yang pada gilirannya akan mencetak informasi biodata mahasiswa ke layar.

12.

```

no usages
1 ▶ public class TestMahasiswa {
    no usages
2 ▶     public static void main(String[] args) {
3         Mahasiswa mhs1 = new Mahasiswa();
4         mhs1.nim=101;
5         mhs1.nama="Lestari";
6         mhs1.alamat="Jl. Vinolia No 1A";
7         mhs1.kelas="1A";
8         mhs1.tampilBiodata();
9
10        Mahasiswa mhs2 = new Mahasiswa();
11        mhs2.nim=101;
12        mhs2.nama="Lestari";
13        mhs2.alamat="Jl. Vinolia No 1A";
14        mhs2.kelas="1A";
15        mhs2.tampilBiodata();
16
17        Mahasiswa mhs3 = new Mahasiswa();
18        mhs3.nim=101;
19        mhs3.nama="Lestari";
20        mhs3.alamat="Jl. Vinolia No 1A";
21        mhs3.kelas="1A";
22        mhs3.tampilBiodata();
23    }
24 }

```

"C:\Program Files\Java\jdk-18.0.2.1\bin

```

Nim      : 101
Nama     : Lestari
Alamat   : Jl. Vinolia No 1A
Kelas   : 1A
Nim      : 101
Nama     : Lestari
Alamat   : Jl. Vinolia No 1A
Kelas   : 1A
Nim      : 101
Nama     : Lestari
Alamat   : Jl. Vinolia No 1A
Kelas   : 1A

```

Process finished with exit code 0

### Percobaan 3: Menulis method yang memiliki argument/parameter dan memiliki return

2.

```
no usages
1 public class Barang {
    1 usage
2     public String namaBrg;
    1 usage
3     public String jenisBrg;
    2 usages
4     public int stok;
5
    no usages
6     public void tampilBarang(){
7         System.out.println("Nama Barang      : " +namaBrg);
8         System.out.println("Jenis Barang      : " +jenisBrg);
9         System.out.println("Stok          : " +stok);
10    }
11
12    //method dengan argumen dan nilai balik (return)
    no usages
13    public int tambahStok(int brgMasuk){
14        int stokBaru = brgMasuk+stok;
15        return stokBaru;
16    }
17 }
```

4.

```

no usages
1 ► public class TestBarang {
    no usages
2 ►     public static void main(String[] args) {
3         Barang brg1 = new Barang();
4         brg1.namaBrg = "Pensil";
5         brg1.jenisBrg = "ATK";
6         brg1.stok = 10;
7         brg1.tampilBarang();
8
9         //menampilkan dan mengisi argumen untuk menambahkan stok barang
10        System.out.println("Stok Baru adalah " +brg1.tambahStok( brgMasuk: 20));
11    }
12 }

```

6.

```

"C:\Program Files\Java\jdk-18.0.2.
Nama Barang      : Pensil
Jenis Barang     : ATK
Stok             : 10
Stok Baru adalah 30

Process finished with exit code 0

```

7. Fungsi argumen dalam suatu method adalah data atau nilai yang dapat diteruskan ke method saat memanggilnya. Argumen ini digunakan oleh method untuk melakukan operasi atau menghasilkan output berdasarkan data yang diberikan saat pemanggilan. Argumen digunakan untuk memberikan input atau informasi yang diperlukan kepada method agar method tersebut dapat berfungsi dengan benar.

8. Kesimpulan tentang penggunaan kata kunci return dalam method:

- return digunakan untuk mengembalikan nilai dari suatu method. Ini berarti method dapat menghasilkan output atau nilai yang dapat digunakan oleh bagian-bagian lain dalam program.
- Suatu method harus memiliki return jika Anda ingin mengembalikan nilai dari method tersebut. Jika method hanya digunakan untuk melakukan tugas tanpa mengembalikan nilai, Anda dapat menggunakan void sebagai tipe kembalian dan tidak perlu menggunakan return.

- return digunakan untuk mengakhiri eksekusi method dan mengirim nilai kembali ke pemanggil method. Setelah return dieksekusi, kontrol akan kembali ke tempat pemanggil method.
- Penggunaan return sangat berguna dalam berbagai kasus, seperti mengembalikan hasil perhitungan matematika, nilai dari operasi string, atau bahkan objek yang dibuat di dalam method.
- Tipe data yang dikembalikan oleh return harus sesuai dengan tipe data yang dideklarasikan dalam tanda kurung siku [] pada deklarasi method. Misalnya, jika method memiliki deklarasi `public int hitungTotal()`, maka Anda harus menggunakan return dengan tipe data `int`.

Jadi, return adalah cara untuk mengembalikan nilai dari method dan digunakan ketika method tersebut harus menghasilkan output yang dapat digunakan oleh bagian lain dalam program.

## Tugas

1. Suatu toko persewaan video game salah satu yang diolah adalah peminjaman, dimana data yang dicatat ketika ada orang yang melakukan peminjaman adalah id, nama member, nama game, dan harga yang harus dibayar. Setiap peminjaman bisa menampilkan data hasil peminjaman dan harga yang harus dibayar. Buatlah class diagram pada studi kasus diatas!

Penjelasan:

- Harga yang harus dibayar diperoleh dari lama sewa x harga.
- Diasumsikan 1x transaksi peminjaman game yang dipinjam hanya 1 game saja.

Jawab:

Peminjaman	Game
<ul style="list-style-type: none"> <li>• id : int</li> <li>• namaMember : String</li> <li>• namaGame : String</li> <li>• hargaTotal : double</li> </ul>	<ul style="list-style-type: none"> <li>• gameId : int</li> <li>• nama : String</li> <li>• harga : double</li> </ul>
<ul style="list-style-type: none"> <li>❖ hitungHarga() : void</li> <li>❖ tampilPeminjaman : void</li> </ul>	

2. Buatlah program dari class diagram yang sudah anda buat di no 1!



```
2 usages
1 public class Game {
    2 usages
2     private int gameID;
    2 usages
3     private String nama;
    2 usages
4     private double harga;
5
    1 usage
6     public Game(int gameID, String nama, double harga) {
7         this.gameID = gameID;
8         this.nama = nama;
9         this.harga = harga;
10    }
11
    no usages
12    public int getGameID() {
13        return gameID;
14    }
15
```

```
no usages
16    public String getNama() {
17        return nama;
18    }
19
    1 usage
20    public double getHarga() {
21        return harga;
22    }
23 }
```

```
2 usages
1 public class Peminjaman {
    2 usages
2     private int id;
    2 usages
3     private String namaMember;
    2 usages
4     private String namaGame;
    2 usages
5     private double hargaTotal;
6
    1 usage
7     public Peminjaman(int id, String namaMember, String namaGame, double lamaSewa, double hargaPerHari) {
8         this.id = id;
9         this.namaMember = namaMember;
10        this.namaGame = namaGame;
11        this.hargaTotal = lamaSewa * hargaPerHari;
12    }
13
    1 usage
14    public void tampilPeminjaman() {
15        System.out.println("ID Peminjaman: " + id);
16        System.out.println("Nama Member: " + namaMember);
17        System.out.println("Nama Game: " + namaGame);
18        System.out.println("Harga Total: " + hargaTotal);
19    }
}

no usages
1 public class MainPeminjaman {
    no usages
2     public static void main(String[] args) {
3         Game game1 = new Game( gameId: 1, nama: "Game A", harga: 5.0);
4         Peminjaman peminjaman1 = new Peminjaman( id: 101, namaMember: "John", namaGame: "Game A", lamaSewa: 3, game1.getHarga());
5
6         peminjaman1.tampilPeminjaman();
7     }
8 }

"C:\Program Files\Java\jdk-18.0.2.1
ID Peminjaman: 101
Nama Member: John
Nama Game: Game A
Harga Total: 15.0

Process finished with exit code 0
```

3. Buatlah program sesuai dengan class diagram berikut ini:

Lingkaran
+phi: double +r: double
+hitungLuas(): double +hitungKeliling(): double

Jawab:

```

2 usages
1  public class Lingkaran {
      2 usages
2      double phi = 3.14;
      5 usages
3      double r;
4
      1 usage
5      double hitungLuas(){
6          double L = phi * r * r;
7          return L;
8      }
9
      1 usage
10     double hitungKeliling(){
11         double K = 2 * phi * r;
12         return K;
13     }
14 }
15 |

```

```

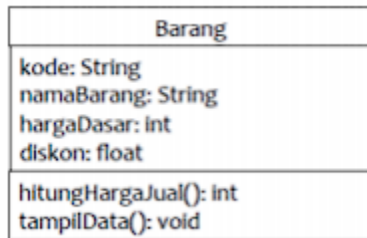
no usages
1  ► public class MainLingkaran {
      no usages
2  ►     public static void main(String[] args) {
3          Lingkaran l1 = new Lingkaran();
4          l1.r = 7;
5          System.out.println("Luas dan permukaan lingkaran dengan r=" + l1.r + " : ");
6          System.out.println("Luas : " + l1.hitungLuas());
7          System.out.println("Keliling : " + l1.hitungKeliling());
8      }
9  }

```

```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java
Luas dan permukaan lingkaran dengan r=7.0 :
Luas : 153.86
Keliling : 43.96

Process finished with exit code 0
```

4. Buatlah program sesuai dengan class diagram berikut ini:



Deskripsi / Penjelasan :

- Nilai atribut hargaDasar dalam Rupiah dan atribut diskon dalam %
- Method hitungHargaJual() digunakan untuk menghitung harga jual dengan perhitungan berikut ini:

harga jual = harga dasar – (diskon x harga dasar)

- Method tampilData() digunakan untuk menampilkan nilai dari kode, namaBarang, hargaDasar, diskon dan harga jual.

Jawab:

```

2 usages
1 public class Barang2 {
    2 usages
2     String kode, namaBarang;
    4 usages
3     int hargaDasar;
    3 usages
4     float diskon;
5
    2 usages
6     int hitungHargaJual(){
7         int hargaJual = (int) (hargaDasar - (diskon/100*hargaDasar));
8         return hargaJual;
9     }
10
    1 usage
11     public void tampilData(){
12         System.out.println("Kode : " +kode);
13         System.out.println("Nama Barang : " +namaBarang);
14         System.out.println("Harga Dasar : Rp." +hargaDasar);
15         System.out.println("Diskon : " +diskon +"%");
16         System.out.println("Harga Jual : Rp. " +hitungHargaJual());
17     }
18 }

```

```

no usages
1 public class MainBarang2 {
    no usages
2     public static void main(String[] args) {
3         Barang2 b2 = new Barang2();
4
5         b2.kode = "01";
6         b2.namaBarang = "Beras";
7         b2.hargaDasar = 30000;
8         b2.diskon = 25;
9         b2.hitungHargaJual();
10        b2.tampilData();
11    }
12 }

```

"C:\Program Files\Java\jdk-18.0.2.1\bin

Kode : 01

Nama Barang : Beras

Harga Dasar : Rp.30000

Diskon : 25.0%

Harga Jual : Rp. 22500

Process finished with exit code 0