

PRAKTIKUM 3 KONSTRUKTOR DAN OVERLOADING

D. PERCOBAAN

1. Melakukan pengaturan format jumlah angka dibelakang koma pada bilangan desimal. Modifikasi main program sebagai pengetesan pada tugas pendahuluan ke-3.

```
1 import java.text.DecimalFormat;
2 public class HitungLingkaranFormatDesimal{
3     public static void main(String args[]){
4         DecimalFormat FormatData = new DecimalFormat("#.##");
5         Lingkaran c = new Lingkaran(2.013);
6         double LuasFormatDesimal;
7         double KelilingFormatDesimal;
8         LuasFormatDesimal = Double.valueOf(FormatData.format(c.getLuas()));
9         KelilingFormatDesimal = Double.valueOf(FormatData.format(c.getKeliling()));
10        System.out.println("Luas = "+LuasFormatDesimal+" dan Keliling "+KelilingFormatDesimal);
11    }
12 }
```

Tampilan yang diinginkan untuk 2 angka dibelakang koma

```
run:
Luas = 12.72 dan Keliling 12.64
BUILD SUCCESSFUL (total time: 0 seconds)
```

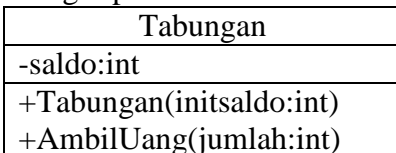
Modifikasi program untuk menampilkan 3, 4 dan 5 angka dibelakang koma

```
run:
Luas = 12.724 dan Keliling 12.642
BUILD SUCCESSFUL (total time: 0 seconds)

run:
Luas = 12.72381 dan Keliling 12.64164
BUILD SUCCESSFUL (total time: 0 seconds)

run:
Luas = 12.7238 dan Keliling 12.6416
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Penggunaan Konstruktor dan Metode
Mengimplementasikan UML class diagram dalam program untuk class Tabungan



Transformasikan class diagram diatas ke dalam bentuk program. Tulis listing program berikut ini sebagai pengetesan.

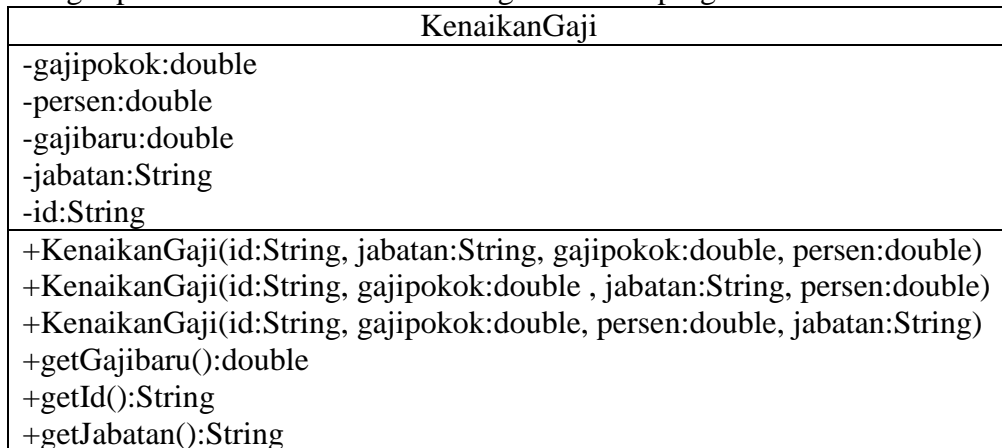
```
10 public class CetakTabungan {
11     public static void main(String args[]){
12         Tabungan tabungan = new Tabungan(1200000);
13         System.out.println("Saldo awal : "+tabungan.saldo);
14         tabungan.AmbilUang(250000);
15         System.out.println("Jumlah uang yang diambil : 250000");
16         System.out.println("Saldo sekarang : "+tabungan.saldo);
17     }
18 }
```

Tampilan yang diharapkan seperti di bawah ini.

```
run:
Saldo awal : 1200000
Jumlah uang yang diambil : 250000
Saldo sekarang : 950000
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Overloading terhadap konstruktor

Mengimplementasikan UML class diagram dalam program untuk class KenaikanGaji.



$$gajibaru = gajipokok + \left(\frac{persen}{100}\right) * gajipokok$$

Transformasikan class diagram diatas ke dalam bentuk program. Tulislah listing program berikut ini sebagai pengetesan.

```
39 public class CetakKenaikanGaji {
40     public static void main(String args[]){
41         KenaikanGaji santi = new KenaikanGaji("SBY00034", "Manager", 4000000, 5);
42         KenaikanGaji ronny = new KenaikanGaji("BDG0567", 2000000, "Staff", 7.5);
43         KenaikanGaji indra = new KenaikanGaji("JKI9876", 8000000, 3.2, "Direktur");
44         System.out.println("Nama : Santi");
45         System.out.println("No Pegawai : "+santi.getId());
46         System.out.println("Jabatan : "+santi.getJabatan());
47         System.out.println("Gaji Baru : "+santi.getGajibaru());
48         System.out.println("");
49
50         System.out.println("Nama : Ronny");
51         System.out.println("No Pegawai : "+ronny.getId());
52         System.out.println("Jabatan : "+ronny.getJabatan());
53         System.out.println("Gaji Baru : "+ronny.getGajibaru());
54         System.out.println("");
55
56         System.out.println("Nama : Indra");
57         System.out.println("No Pegawai : "+indra.getId());
58         System.out.println("Jabatan : "+indra.getJabatan());
59         System.out.println("Gaji Baru : "+indra.getGajibaru());
60         System.out.println("");
61     }
62 }
```

Tampilan yang diharapkan seperti di bawah ini.

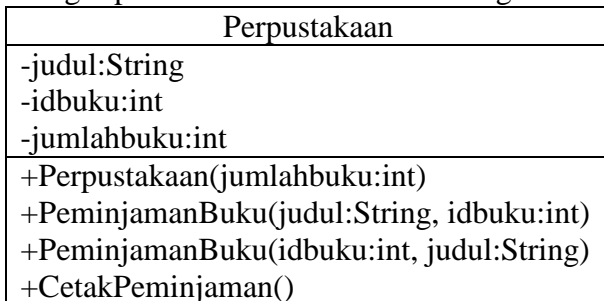
```
run:
Nama : Santi
No Pegawai : SBY00034
Jabatan : Manager
Gaji Baru : 4200000.0

Nama : Ronny
No Pegawai : BDG0567
Jabatan : Staff
Gaji Baru : 2150000.0

Nama : Indra
No Pegawai : JKT9876
Jabatan : Direktur
Gaji Baru : 8256000.0

BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Overloading terhadap metode
Mengimplementasikan UML class diagram dalam program untuk class Perpustakaan



Transformasikan class diagram diatas ke dalam bentuk program. Tulislah listing program berikut ini sebagai pengetesan.

```
23 public class CetakPerpustakaan {
24     public static void main(String args[]){
25         Perpustakaan anggota = new Perpustakaan(0);
26         System.out.println("=====Daftar Peminjaman Buku=====");
27         anggota.PeminjamanBuku("Dasar Pemrograman Java 2", 12098);
28         anggota.CetakPeminjaman();
29         anggota.PeminjamanBuku(23337, "Pemrograman Berorientasi Objek");
30         anggota.CetakPeminjaman();
31         System.out.println("Jumlah Buku : "+anggota.jumlahbuku);
32         System.out.println("=====");
33     }
34 }
```

Tampilan yang diharapkan seperti di bawah ini.

```
run:
=====Daftar Peminjaman Buku=====
Judul Buku : Dasar Pemrograman Java 2
ID Buku : 12098
Judul Buku : Pemrograman Berorientasi Objek
ID Buku : 23337
Jumlah Buku : 2
=====
BUILD SUCCESSFUL (total time: 0 seconds)
```

E. TUGAS RESMI

1. Analisa dan berikan kesimpulan pada percobaan yang sudah dilakukan
2. Jelaskan apa yang dimaksud dengan overloading pada konstruktor dan metode ?
3. Buat UML class diagram yang di dalamnya digunakan konstruktor dan metode serta implementasikan dalam bentuk program. Selanjutnya buat main program untuk melakukan pengetesan dari program yang sudah dibuat.
4. Buat UML class diagram yang di dalamnya digunakan overloading terhadap konstruktor atau overloading terhadap metode (pilih salah satu), serta implementasikan dalam bentuk program. Selanjutnya buat main program untuk melakukan pengetesan dari program yang sudah dibuat.