PERTEMUAN KEEMPAT

Polymorphism, Class Association

Prepared by Adi Wahyu Pribadi

Polymorphism

Polimorfisme dalam OOP adalah sebuah prinsip di mana class dapat memiliki banyak bentuk method yang berbeda-beda meskipun namanya sama. “Bentuk” di sini dapat kita artikan: isinya berbeda, parameternya berbeda, dan tipe datanya berbeda.

Polimorfisme pada Java ada dua macam:

1. Static Polymorphism (Polimorfisme statis);
2. Dynamic Polymorphism (Polimorfisme dinamis).

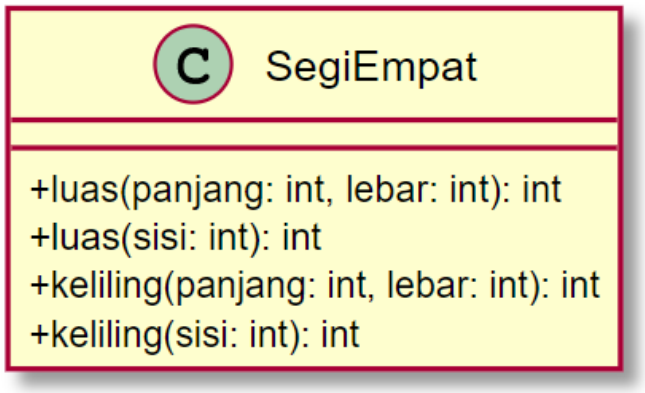
Polimorfisme statis menggunakan **method overloading** sedangkan polimorfisme dinamis menggunakan **method overriding**.

Static Polymorphism/Method Overloading

Method overloading terjadi pada sebuah class yang memiliki nama method yang sama tapi memiliki parameter dan tipe data yang berbeda.

Sebagai contoh ada dua bangun datar segi empat yaitu segi empat yang memiliki panjang dan lebar berbeda, dan segi empat yang panjang dan lebarnya sama atau biasa disebut bujur sangkar. Menghitung luas dan keliling kedua bangun datar tersebut kita gunakan rumus berbeda. Maka method luas dan keliling memiliki paramater yang berbeda.

Parameter input luas dan keliling segi empat adalah panjang dan lebar, dan parameter input bujur sangkar adalah sisi. Sebagaimana terlihat pada berikut yaitu class diagram SegiEmpat yang memiliki 4 method. Ada dua method untuk menghitung luas karena luas **SEGI EMPAT** adalah ***panjang x lebar***, dan luas **BUJUR SANGKAR** adalah ***sisi x sisi***. Sedangkan method untuk menghitung keliling juga ada dua. Keliling **SEGI EMPAT** adalah ***(panjang + lebar) x 2*** dan keliling **BUJUR SANGKAR** adalah ***sisi x 4***.

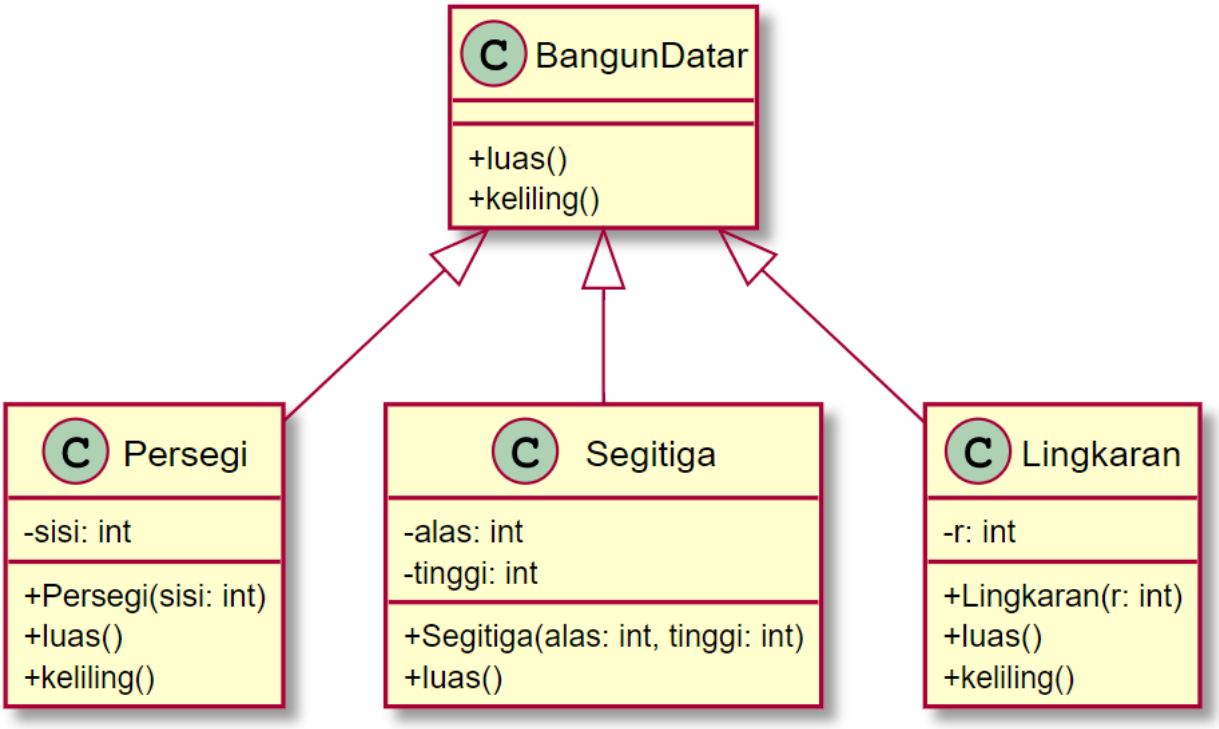



```
class SegiEmpat {
    public int luas(int panjang, int lebar) {
        return panjang * lebar;
    }

    public int luas(int sisi) {
        return sisi * 4;
    }

    public int keliling(int panjang, int lebar) {
        return (panjang + lebar) * 2;
    }

    public int keliling(int sisi) {
        return sisi * 4;
    }
}


public class App {
    public static void main(String[] args) throws Exception {
        SegiEmpat s1 = new SegiEmpat();
        SegiEmpat bs = new SegiEmpat();

        System.out.println("Luas Segi Empat: " + s1.luas(10, 4));
        System.out.println("Luas Bujursangkar: " + bs.luas(8));
        System.out.println("Keliling Segi Empat: " + s1.keliling(10, 4));
        System.out.println("Keliling Bujur Sangkar: " + bs.keliling(8));
    }
}
```

Output:

```
Luas Segi Empat: 40
Luas Bujursangkar: 32
Keliling Segi Empat: 28
Keliling Bujur Sangkar: 32
```

Dynamic Polymorphism/Method Overriding



Pada diagram tersebut, terdapat class BangunDatar yang memiliki tiga subclass, yaitu: Persegi, Lingkaran, dan Segitiga.

Setiap class memiliki method yang sama yaitu luas() dan keliling(). Akan tetapi method-method ini memiliki isi rumus yang berbeda.

Berikut Source Code dari class diagram di atas yaitu BangunDatar.java, Persegi.java, Segitiga.java dan Lingkaran.java

BangunDatar.java

```
package app;
```

```java
public class BangunDatar {
    float luas(){
        System.out.println("Menghitung luas bangun datar");
        return 0;
    }

    float keliling(){
        System.out.println("Menghitung keliling bangun datar");
        return 0;
    }
}
```

Persegi.java

```java
package app;

class Persegi extends BangunDatar{
    private int sisi;

    public Persegi(int sisi) {
        this.sisi = sisi;
    }

    @Override
    public float luas() {
        return this.sisi * this.sisi;
    }

    @Override
    public float keliling(){
        return this.sisi * 4;
    }
}
```

Segitiga.java

```java
package app;

class Segitiga extends BangunDatar{
    private int alas;
    private int tinggi;

    public Segitiga(int alas, int tinggi) {
        this.alas = alas;
        this.tinggi = tinggi;
    }

    @Override
    public float luas(){
        return this.alas * this.tinggi;
    }
 }
```

Lingkaran.java

```java
package app;

class Lingkaran extends BangunDatar {
    private int r;

    public Lingkaran(int r) {
        this.r = r;
    }

    @Override
    public float luas(){
        return (float) (Math.PI * r * r);
    }

    @Override
    public float keliling(){
        return (float) (2 * Math.PI * r);
    }

}
```

App.java

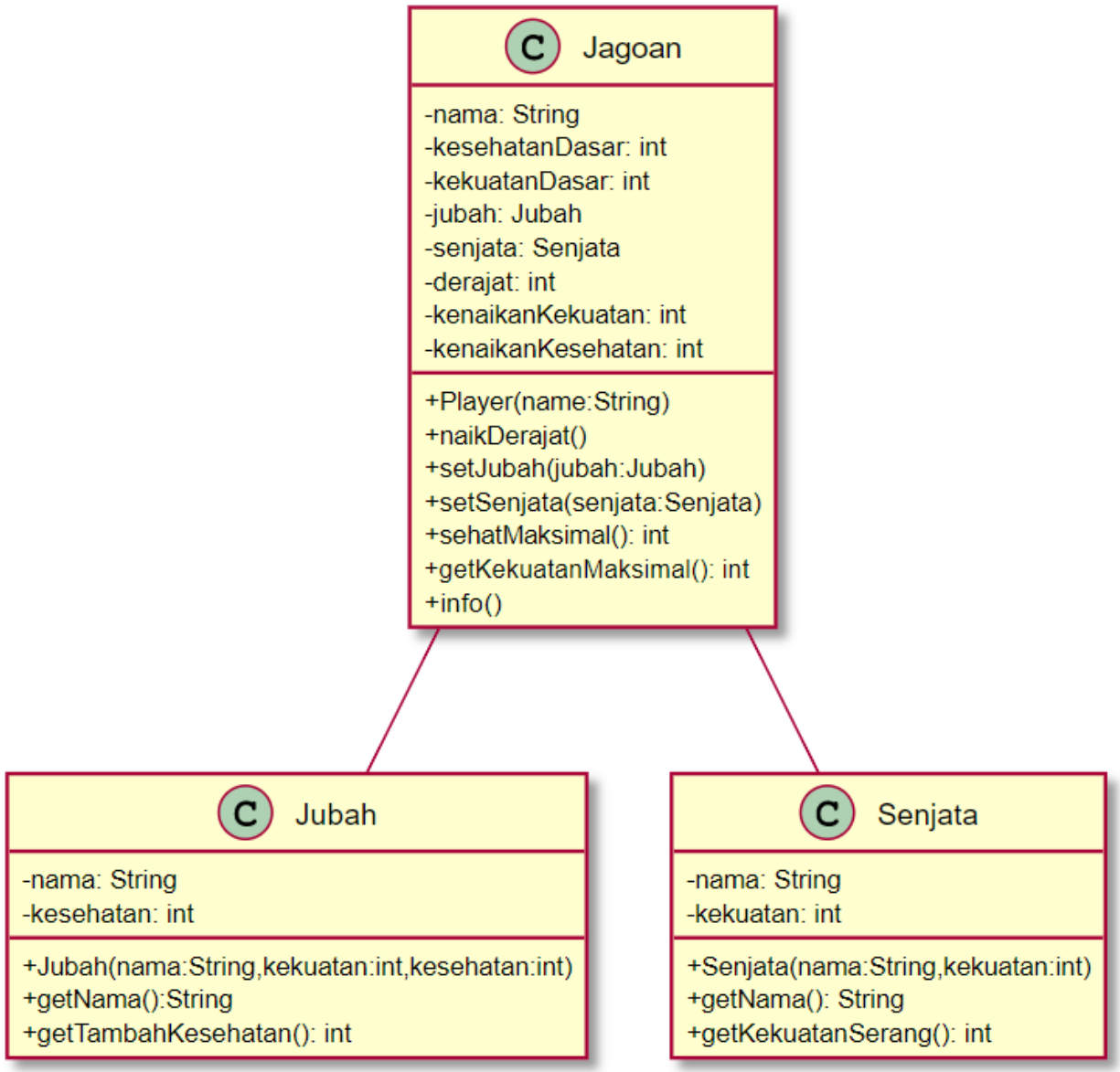
```
package app;

public class App {
    public static void main(String[] args) throws Exception {
        BangunDatar bangunDatar = new BangunDatar();
        Persegi persegi = new Persegi(4);
        Segitiga segitiga = new Segitiga(6, 3);
        Lingkaran lingkaran = new Lingkaran(50);

        // memanggil method luas dan keliling
        bangunDatar.luas();
        bangunDatar.keliling();
        System.out.print("\n");
        System.out.println("Luas persegi: " + persegi.luas());
        System.out.println("keliling persegi: " + persegi.keliling());
        System.out.print("\n");
        System.out.println("Luas segitiga: " + segitiga.luas());
        System.out.print("\n");
        System.out.println("Luas lingkaran: " + lingkaran.luas());
        System.out.println("keliling lingkaran: " + lingkaran.keliling());
    }
}
```

Output:

```
Menghitung luas bangun datar
Menghitung keliling bangun datar

Luas persegi: 16.0
keliling persegi: 16.0

Luas segitiga: 18.0

Luas lingkaran: 7853.9814
keliling lingkaran: 314.15927
```

Studi Kasus: Games Jagoan

Class Diagram



Pada Class Diagram di atas, terlihat class Jagoan, class Jubah, dan class Senjata. Biasa pada Games Pertempuran, seorang jagoan memiliki jubah dan senjata andalannya. Bisa ada banyak jagoan, banyak jubah, dan banyak senjata dalam suatu pertempuran.

Berikut adalah Source Code

Jagoan.java

```java
package app;

class Jagoan {
    private String nama;
    private int kesehatanDasar;
    private int kekuatanDasar;
    private Jubah jubah; // asosiasi
    private Senjata senjata; // asosiasi
    private int derajat;
    // private int kenaikanKekuatan;
    // private int kenaikanKesehatan;

    public Jagoan(String nama) {
        this.nama = nama;
        this.kesehatanDasar = 100;
        this.kekuatanDasar = 100;
        this.derajat = 1;
        // this.kenaikanKekuatan = 20;
        // this.kenaikanKesehatan = 20;
    }

    public void naikDerajat() {
        this.derajat++;
    }

    public void setJubah(Jubah jubah) {
        this.jubah = jubah;
    }

    public void setSenjata(Senjata senjata) {
        this.senjata = senjata;
    }

    public int sehatMaksimal() {
        return this.kesehatanDasar + this.jubah.getTambahKesehatan();
    }

    public int getKekuatanMaksimal() {
        return this.kekuatanDasar + this.senjata.getKekuatanSerang();
    }
```

```
    public void info() {
        System.out.println("Jagoan\t\t\t: " + this.nama);
        System.out.println("Derajat\t\t\t: " + this.derajat);
        System.out.println("Kesehatan Dasar\t\t: " + this.kesehatanDasar);
        System.out.println("Kekuatan Dasar\t\t: " + this.kekuatanDasar);
        System.out.println("Jubah\t\t\t: " + this.jubah.getNama());
        System.out.println("Senjata\t\t\t: " + this.senjata.getNama());
        System.out.println("Kesehatan Maksimal\t: " +
this.sehatMaksimal());
        System.out.println("Kekuatan Maksimal\t: " +
this.getKekuatanMaksimal() + "\n");
    }
}
```

Jubah.java

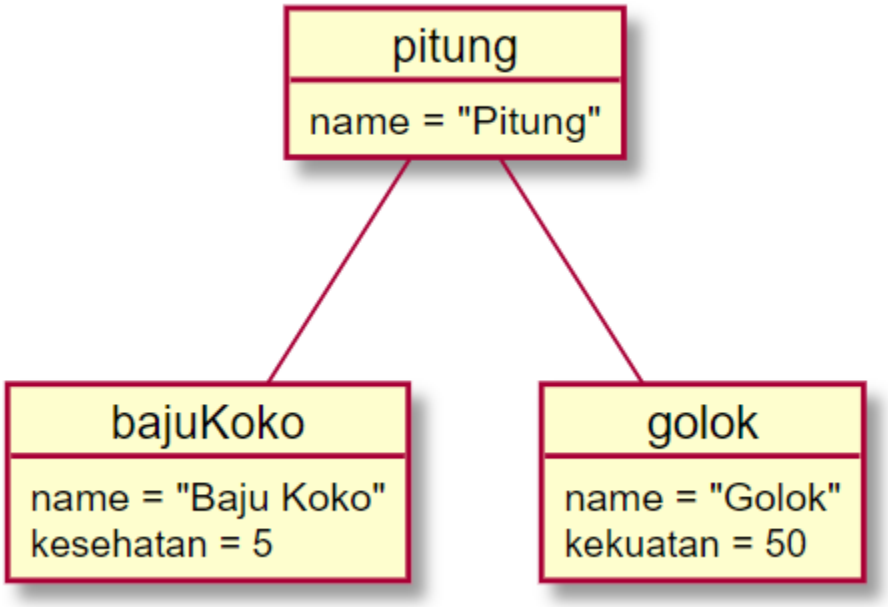
```
package app;

class Jubah {
    private String nama;
    private int kesehatan;

    public Jubah(String nama, int kesehatan) {
        this.nama = nama;
        this.kesehatan = kesehatan;
    }

    public String getNama() {
        return this.nama;
    }

    public int getTambahKesehatan() {
        return this.kesehatan * 10;
    }
}
```

Senjata.java

```java
package app;

class Senjata {
    private String nama;
    private int kekuatan;

    public Senjata(String nama, int kekuatan) {
        this.nama = nama;
        this.kekuatan = kekuatan;
    }

    public String getNama() {
        return this.nama;
    }

    public int getKekuatanSerang() {
        return this.kekuatan * 2;
    }
}
```

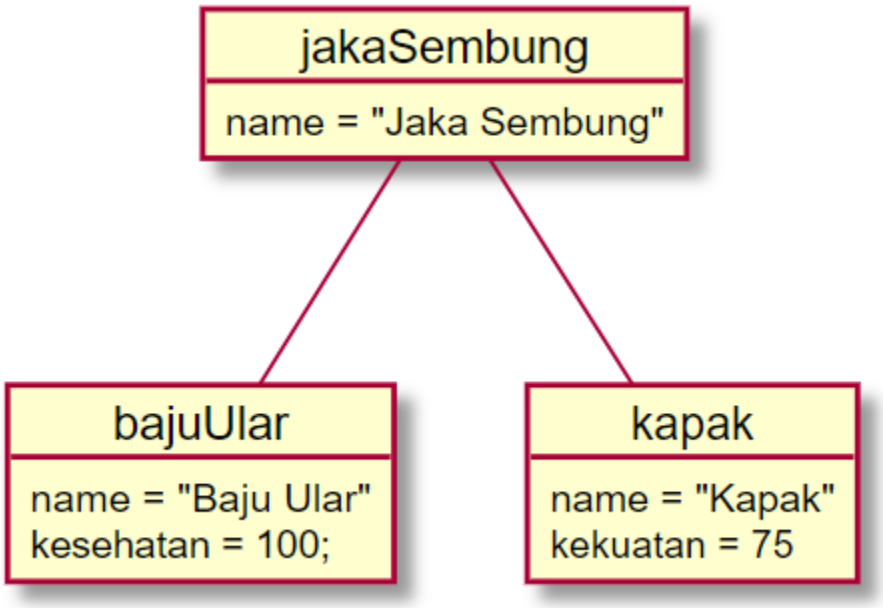
Object Diagram UML

Pitung



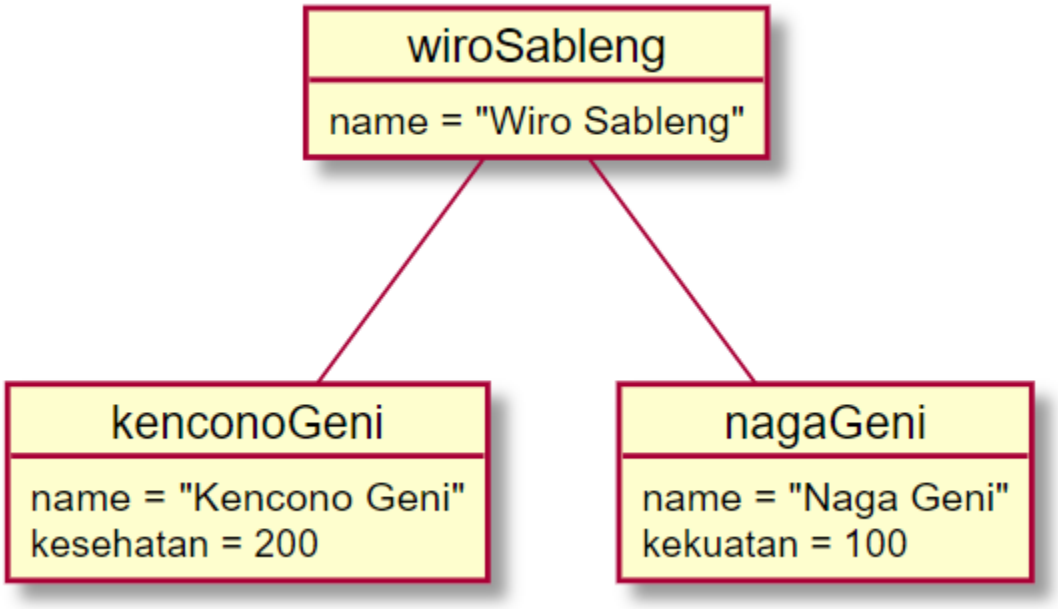

```
public class App {
    public static void main(String[] args) {
        Jagoan pitung = new Jagoan("Pitung");
        Jubah bajuKoko = new Jubah("Baju Koko", 5);
        Senjata golok = new Senjata("Golok", 50);

        pitung.setJubah(bajuKoko);
        pitung.setSenjata(golok);

        pitung.info();
    }
}
```

Jaka Sembung




```
public static void main(String[] args) {
        Jagoan jakaSembung = new Jagoan("Jaka Sembung");
        Jubah bajuUlar = new Jubah("Baju Ular", 100);
        Senjata kapak = new Senjata("Kapak", 75);

        jakaSembung.setJubah(bajuUlar);
        jakaSembung.setSenjata(kapak);

        jakaSembung.info();
}
```

Wiro Sableng




```
public static void main(String[] args) {
        Jagoan wiroSableng = new Jagoan("Wiro Sableng");
        Jubah kenconoGeni = new Jubah("Kencono Geni", 200);
        Senjata nagaGeni = new Senjata("Kapak Naga Geni", 100);

        wiroSableng.setJubah(kenconoGeni);
        wiroSableng.setSenjata(nagaGeni);

        wiroSableng.info();
}
```