

# Konsep Lanjut Pemrograman Berorientasi Objek

---



# Polymorphism

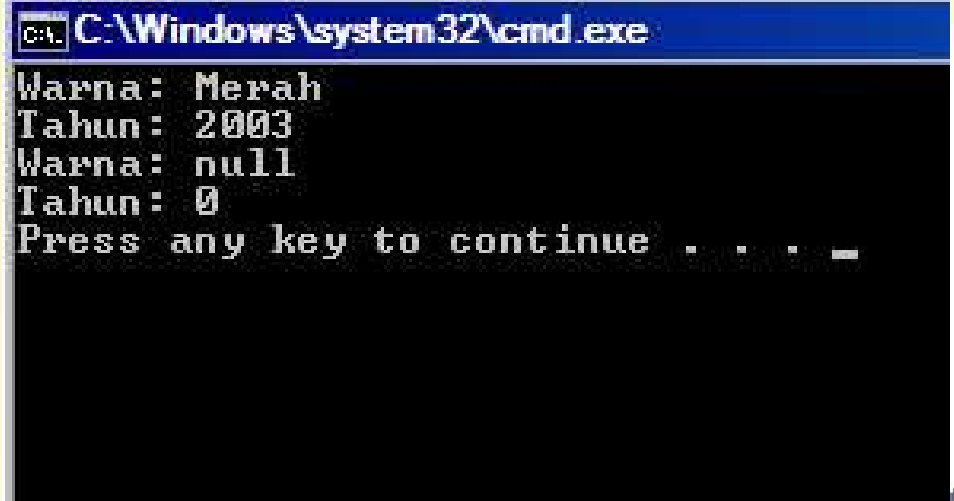
---

- **Polymorphism**: Suatu object dapat memiliki berbagai bentuk, sebagai object dari classnya sendiri atau object dari superclassnya
  1. **Overloading**: Penggunaan **satu nama** untuk beberapa **method yang berbeda parameter**
  2. **Overriding**: Terjadi ketika deklarasi method subclass **sama dengan** method dari superclassnya

# Polymorphism – Overloading

```
class Mobil {  
    String warna;  
    int tahunProduksi;  
  
    public Mobil(String warna, int  
        tahunProduksi){  
        this.warna = warna;  
        this.tahunProduksi = tahunProduksi;  
    }  
  
    public Mobil(){  
    }  
  
    void info(){  
        System.out.println("Warna: " +  
            this.warna);  
        System.out.println("Tahun: " +  
            this.tahunProduksi);  
    }  
}
```

```
public class MobilKonstruktor{  
    public static void main(String[] args){  
        Mobil mobilku = new Mobil("Merah",  
            2003);  
        mobilku.info();  
  
        Mobil mobilmu = new Mobil();  
        mobilmu.info();  
    }  
}
```



```
C:\Windows\system32\cmd.exe  
Warna: Merah  
Tahun: 2003  
Warna: null  
Tahun: 0  
Press any key to continue . . .
```

# Contoh Lain Overloading

---

```
class Lingkaran{  
    public buatLingkaran(int diameter){  
        ...  
    }  
    public buatLingkaran(int diameter, int x, int y){  
        ...  
    }  
    public buatLingkaran(int diameter, int x, int y, int  
        warna, String namaLingkaran){  
        ...  
    }  
}
```

# Polymorphism - Overriding

```
class Sepeda{  
    int kecepatan = 0;  
    int gir = 0;  
  
    // method  
    void ubahGir(int pertambahanGir) {  
        gir= gir+ pertambahanGir;  
        System.out.println("Gir:" + gir);  
    }  
    void tambahKecepatan(int pertambahanKecepatan) {  
        kecepatan = kecepatan+ pertambahanKecepatan;  
        System.out.println("Kecepatan:" + kecepatan);  
    }  
}
```



# Class SepedaGunung Mewarisi Class Sepeda

```
class SepedaGunung extends Sepeda {  
  
    int setSadel;  
  
    void ubahGir(int pertambahanGir)  
    {  
        gir= 2*(gir+ pertambahanGir );  
        System.out.println("Gir:" + gir);  
    }  
}
```

SepedaGunung.java

```
class SepedaGunungBeraksi {  
    public static void main(String[] args) {  
        // Membuat object  
        SepedaGunung sepedaku = new SepedaGunung();  
  
        // Memanggil method di object  
        sepedaku.tambahKecepatan(10);  
        sepedaku.ubahGir(2);  
        sepedaku.setSadel(20);  
  
    }  
}
```

SepedaGunungBeraksi.java

# Latihan

---

1. Kembangkan class **Matematika** dan **MatematikaBeraksi**
2. Lakukan **overloading** pada **Method** yang ada (pertambahan, pengurangan, perkalian, pembagian) . Method baru adalah bertipe data **double** (pecahan) dan **memiliki 3 parameter**
3. Uji di kelas **MatematikaBeraksi** dengan parameter pecahan: 12.5, 28.7, 14.2
4. Misalnya:  
pertambahan(12.5, 28.7, 14.2)

# Matematika.java

---

```
class Matematika{  
    int hasilPertambahanBulat;  
    double hasilPertambahanPecahan;  
  
    public void pertambahan (int a, int b){  
        hasilPertambahanBulat = a + b;  
        System.out.println("hasil:" + hasilPertambahanBulat);  
    }  
  
    public void pertambahan (double a, double b, double c){  
        hasilPertambahanPecahan = a + b + c;  
        System.out.println("hasil:" + hasilPertambahanPecahan);  
    }  
}
```

...



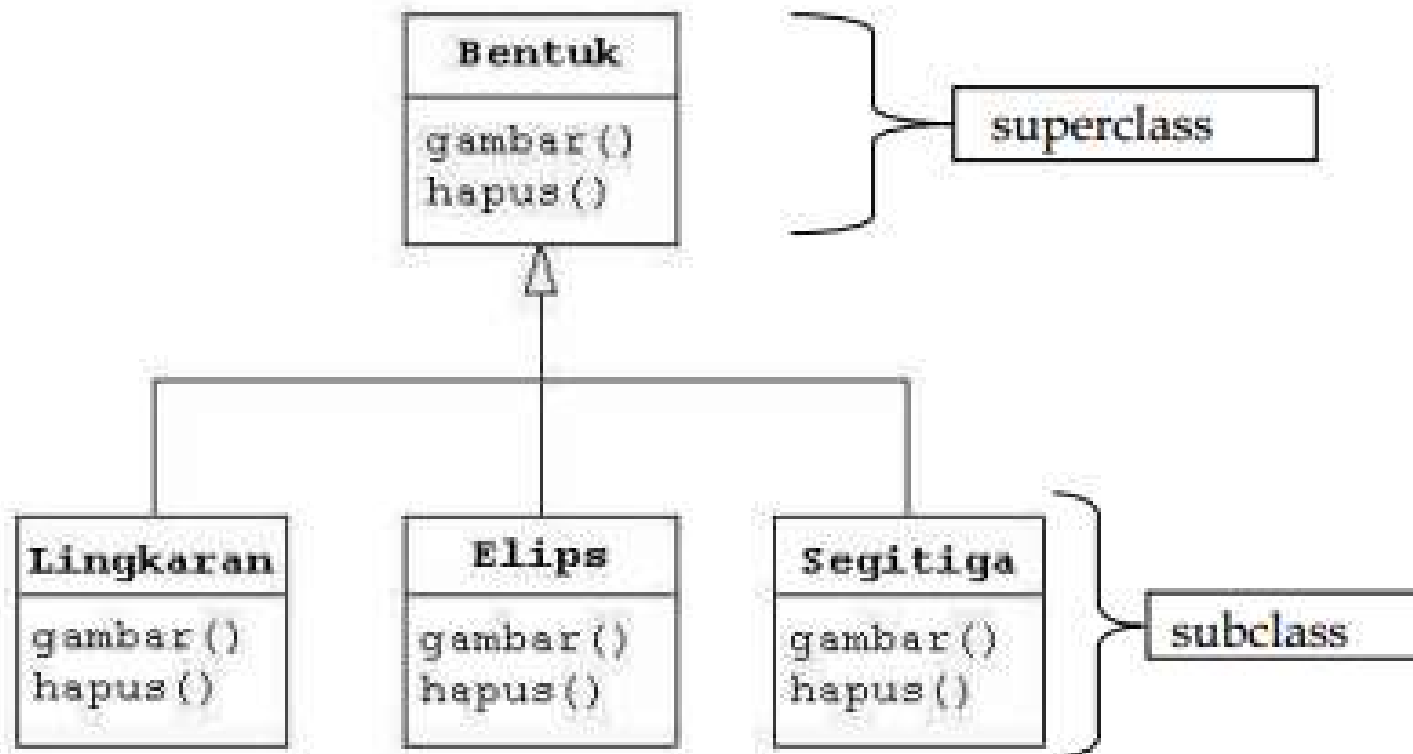
# Tugas:

---

1. Kembangkan class **Buku** dan BukuDemo
2. Selain konstruktor class Buku dengan parameter: **Judul, Pengarang, Penerbit dan Tahun**, buat konstruktor lagi dengan parameter: **Judul, Pengarang** saja
3. Panggil kedua konstruktor class **Buku** tadi dari class **BukuDemo**

# Override

---



# Override

```
class Bentuk
{
    public void gambar() {
        System.out.println("Menggambar");
    }
    public void hapus() {
        System.out.println("Menghapus
Gambar");
    }
}

class Lingkaran extends Bentuk
{
    public void gambar(){
        System.out.println("Gambar
Lingkaran");
    }
    public void hapus() {
        System.out.println("Hapus
Lingkaran");
    }
}
```

```
class Elips extends Bentuk
{
    public void gambar() {
        System.out.println("Gambar Elips");
    }
    public void hapus() {
        System.out.println("Hapus Elips");
    }
}

class Segitiga extends Bentuk
{
    public void gambar() {
        System.out.println("Gambar
Segitiga");
    }
    public void hapus() {
        System.out.println("Hapus
Segitiga");
    }
}
```



# Override

---

```
class RandomBentuk
{
    private Random rand = new Random();
    public Bentuk next() {
        switch(rand.nextInt(3))
        {
            default:
            case 0: return new Lingkaran();
            case 1: return new Elips();
            case 2: return new Segitiga();
        }
    }
}

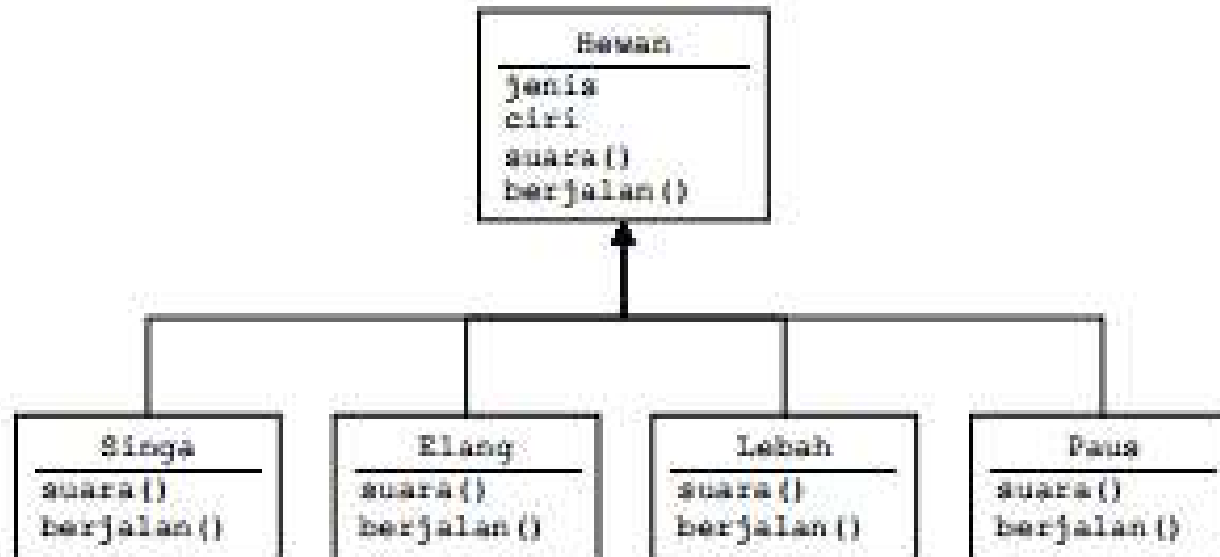
class Latihan3a
{
    private static RandomBentuk gen = new
        RandomBentuk();
    public static void main(String[] args)
    {
        Bentuk[] bangun = new Bentuk[3];
        for(int i = 0; i < bangun.length;
i++)
            bangun[i] = gen.next();
        for(int i = 0; i < bangun.length;
i++)
            bangun [i].gambar();
        for(int i = 0; i < bangun.length;
i++)
            bangun [i].hapus();
    }
}
```



# Override

## Latihan

- 1 Terdapat class-class hewan sebagai berikut :



Penjelasan dari masing-masing atribut dan method sebagai berikut :

`jenis` : apakah termasuk hewan mamalia, serangga atau burung

`ciri` : ciri dari hewan tersebut

`suara()` : suara hewan tersebut

`berjalan()` : cara hewan tersebut berjalan/bergerak

# Rehat Sejenak

---

- ZOO