



Pewarisan(Inheritance)

Oleh:
Mike Yuliana
PENS-ITS

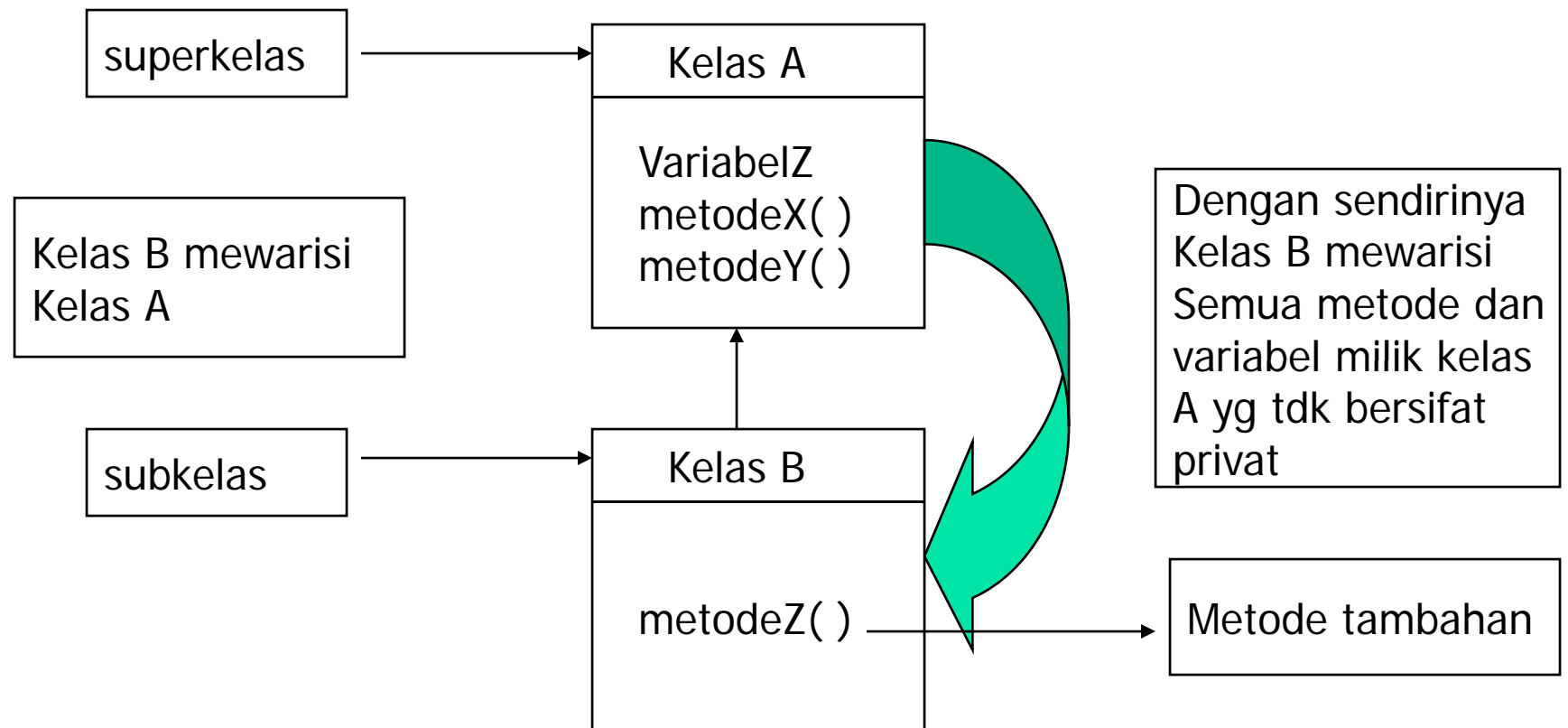


Tujuan

- Pengertian Pewarisan
- Cara mewariskan kelas
- Penentu akses protected
- Memanggil metode Superkelas
- Memanggil konstruktor superkelas
- Penentu akses final
- Penentu abstract
- Polymorphisme

Pengertian Pewarisan

Pewarisan merupakan konsep dalam pemrograman berorientasi objek yang memungkinkan untuk membuat suatu kelas dengan didasarkan pada kelas yang sudah ada sehingga mewarisi semua metode dan variabelnya





Cara mewariskan kelas

1. Pembuatan superkelas alpha.java

```
class alpha
{
    private int a=5;

    public void info( )
    {
        System.out.println("a="+a);
    }
}
```



2. Pembuatan subkelas bravo.java

bentuk:

```
class subkelas extends superkelas  
{  
    tubuh kelas;  
}
```

Bravo.java

```
class bravo extends alpha  
{  
    private int b;  
    public void infolagi()  
    {  
        System.out.println("b="+b);  
    }  
}
```



3. Kompilasi program dibawah ini

```
public class coba
{
    public static void main(String args[])
    {
        alpha ob=new alpha();
        bravo obj=new bravo();
        obj.info();
        ob.infolagi();
    }
}
```

Error!

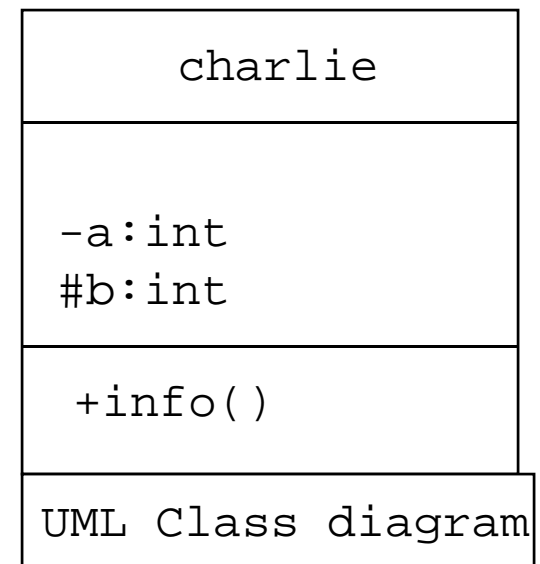



Penentu Akses protected

Penentu akses ini berguna untuk mengatur agar suatu variabel ataupun metode yang terdapat pada sebuah kelas dapat diakses pada kelas itu sendiri dan subkelas

```
class charlie
{
    private int a=5;
    protected int b=7;

    public void info()
    {
        System.out.println("a= "+a);
        System.out.println("b= "+b);
    }
}
```





```
class delta extends charlie
{
    private int c=9;

    public void info()
    {
        System.out.println("a="+a);
        System.out.println("b="+b);
        System.out.println("c="+c);
    }
}
```

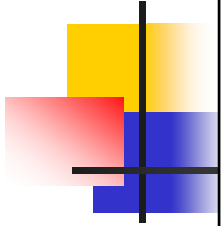
Hasil:

I:\modulajar\dasarpemrograman\moduljava09\cobaku\src\cobaku.java:29: a has private access in charlie

`System.out.println("a="+a);`

1 error

BUILD FAILED (total time: 0 seconds)|



```
class penentuprotect
{
    public static void main(String args[])
    {
        charlie objx=new charlie();
        objx.info();

        delta objy=new delta();
        objy.info();
    }
}
```

Hasil:

a=5

b=7

b=7

c=9

Memanggil Metode Superkelas

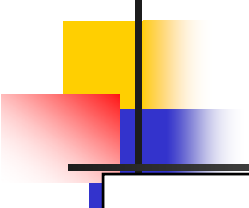
```
class charlie
{
    private int a=5;
    protected int b=7;

    public void info()
    {
        System.out.println("a= "+a);
        System.out.println("b= "+b);
    }
}
```

Metode superkelas

```
class echo extends charlie
{
    private int c=9;

    public void info()
    {
        super.info();
        System.out.println("c= "+c);
    }
}
```



```
class superku
{
    public static void main(String args[])
    {

        echo obj=new echo();
        obj.info();
    }
}
```

Hasil

a=5

b=7

C=9



Memanggil Konstruktor Superkelas

Cara pemanggilan konstruktor oleh subkelas:

```
super ( argumen1 , argumen2 , ... )
```

Cara pemanggilan konstruktor superkelas:

- pemanggilan konstruktor superkelas harus berkedudukan sebagai pernyataan pertama dalam konstruktor

Memanggil konstruktor superkelas

Cara pemanggilan konstruktor oleh subkelas:

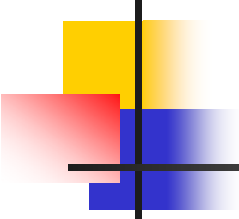
```
super ( argumen1 , argumen2 , ... )
```

```
class orang
{
    private String nama;
    private int usia;
    public orang(String nama,int usia)
    {
        this.nama=nama;
        this.usia=usia;
    }
    public void info()
    {
        System.out.println("nama:" +
            this.nama);
        System.out.println("usia:" +
            this.usia);
    }
}
```

```
class pegawai extends orang
{
    protected String nopeg;

    public pegawai(String nopeg
        String nama,int usia)
    {
        super(nama,usia);
        this.nopeg=nopeg;
    }

    public void info()
    {
        System.out.println("nopeg:" +
            this.nopeg);
        super.info();
    }
}
```



```
public class konstruktorsuperkelas
{
    public static void main(String args[])
    {
        pegawai programmer=new pegawai("123456","edi",23);
        programmer.info();
    }
}
```

Hasil:

nopeg:123456

nama:edi

usia:23



Penentu Akses Final

penentu akses final dapat diterapkan pada variabel atau metode.

- Pada variabel, final akan membuat variabel menjadi suatu konstanta (tidak dapat diubah)
- pada metode, final akan membuat kelas tidak dapat dioverride pada subkelas

```
class abc
{
    final public double PI=3.14;

    final void info()
    {
        System.out.println("PI:"+PI);
    }
}
```

```
class xyz extends abc
{
    public void info()
    {
    }
}
```

akan menimbulkan kesalahan karena
info() tidak dapat dioverride

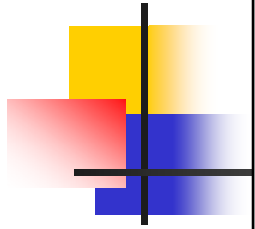
```
class penentu
{
    public static void main(String args[])
    {
        xyz obj=new xyz();
        obj.info();
    }
}
```




Penentu Abstract

Java menyediakan suatu mekanisme yang memungkinkan Metode ditentukan dari dalam kelas tetapi tidak disertai Dengan definisinya. Metode ini dikenal dengan nama metode Abstrak, sedangkan kelasnya dinamakan kelas abstrak

```
public abstract class kendaraan
{
    protected String nama;
    public abstract void jalankan();
}
```



```
class mobil extends kendaraan
{
    public mobil(String nama)
    {
        this.nama=nama;
    }
    public void jalankan()
    {
        System.out.println("nama:"+this.nama+"cara
        menjalankan distater mesinnya");
    }
}
```

tambahkan kelas sepeda!!

```
public class tes
{
    public static void main(String args[]
    {
        mobil mobilku=new mobil("sedan");
        mobilku.jalankan();
    }
}
```

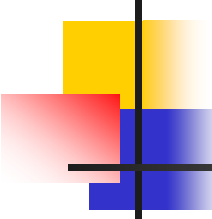
Hasil:
nama:sedan cara menjalankan
distater mesinnya



Polymorphisme

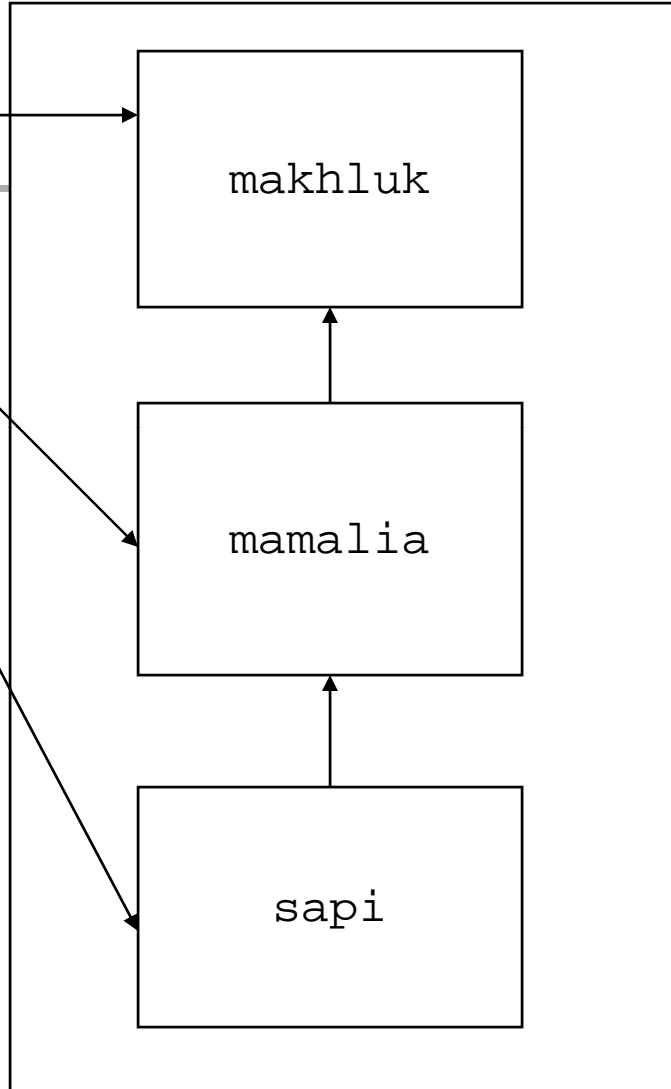
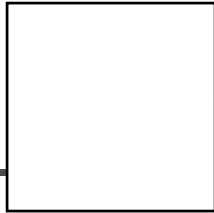
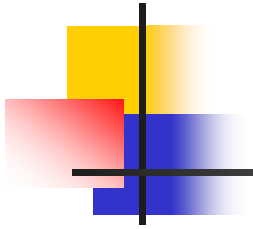
Jika kita mempunyai variabel objek yang bertipe superkelas variabel objek ini bisa diisi dengan objek superkelas atau subkelas tanpa memerlukan perubahan tipe. Model ini dikenal dengan nama polimorfisme, yaitu kemampuan untuk berhubungan dengan berbagai tipe didasarkan pada fitur yang sama

```
class makhluk
{
    public void info()
    {
        System.out.println("info makhluk");
    }
}
class mamalia extends makhluk
{
    public void info()
    {
        System.out.println("info mamalia");
    }
}
class sapi extends mamalia
{
    public void info()
    {
        System.out.println("info sapi");
    }
}
```



```
public class polimorfisme
{
    public static void main(String args[])
    {
        makhluk binatang=new makhluk();
        mamalia mamaliaku=new mamalia();
        sapi sapisumba=new sapi();
        binatang=mamaliaku;
        binatang.info();
        binatang=sapisumba;
        binatang.info();
    }
}
```

binatang



variabel objek yang berkelas superkelas dapat menunjuk ke Subkelas atau sub-subkelas