INFORMATION HIDING, ENCAPSULATION, INHERITANCE, DAN POLYMORHISM

Information Hiding dan Encapsulation

- **Information Hiding** adalah **menyembunyikan attribute** dan method suatu objek dari objek lain.
- **Encapsulation** adalah **menyembunyikan attribute** suatu objek dari objek lain.
- Attribute maupun method disembunyikan dengan cara memberikan modifier **private**.

Information Hiding dan Encapsulation

- method **setter** : method dalam kelas yang sama, yang **memberikan nilai** pada attribute private
- method **getter** : method masih dalam kelas yang sama, yang **mengambil nilai** dari attribute private

contoh program Information Hiding dan Encapsulation

PersegiPanjang.java

```
public class PersegiPanjang{
   private double panjang; // attribute yg di hide
   private double lebar;      // attribute yg di hide
   private double tinggi;     // attribute yg di hide
   public PersegiPanjang() {
      panjang = 0;
      lebar = 0;
   }
   private double luas(double p, double l) {   // di encap
      return p*l;
   }
   public void setPanjang(double panjang) {
      this.panjang = panjang;
   }
   public void setLebar(double lebar) {
      this.lebar = lebar;
   }
   public double getPanjang() {
         return panjang;
      }
      public double getLebar() {
         return lebar;
      }
      public double getLuas() {
         return luas(panjang, lebar);
      }
}
```

MainPersegiPanjang.java

```
public  class MainPersegiPanjang {
   public static void main(String[] srgs) {
      PersegiPanjang pp = new PersegiPanjang();
      pp.setPanjang(10);
      pp.setLebar(20);
      System.out.println("Panjang : "+ pp.getPanjang());
      System.out.println("Lebar : "+ pp.getLebar());
      System.out.println("Luas : "+ pp.getLuas());
   }
}
```

Output

```
Panjang : 10.0
Lebar : 20.0
Luas : 200.0
Press any key to continue . . .
```

Inheritance

- Semua attribute dan method dari suatu kelas super dapat diwariskan ke subkelas
- Bentuk pewarisan :

```
[modifier] class namaSubKelas extend
    namaKelasSuper {
// classBody
}
```

Overrriding Attribute dan Method

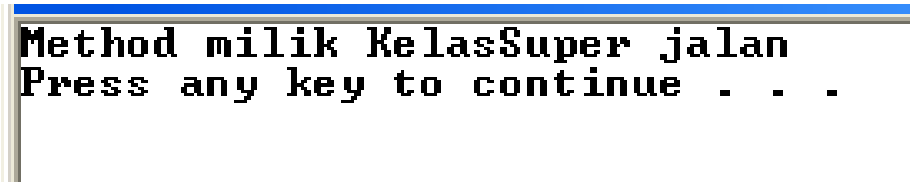
- Overriding adalah kemampuan suatu subkelas untuk memodifikasi attribute dan method milik kelas supernya (bukan private atau final).
- Modifikasi : jumlah parameter, tipe parameter, tipe return value, ataupun lingkungan pemrosesan datanya

contoh program overriding :

KelasSuper.java

```
class KelasSuper {
   public void methodAsli() {
      System.out.println("Method milik
         KelasSuper jalan");
   }
   public static void main(String[] args) {
      KelasSuper oks = new KelasSuper();
      oks.methodAsli();
   }
}
```

Output



contoh program overriding :

SubKelas.java

```
class SubKelas extends KelasSuper {
    public void methodAsli() {
       System.out.println("Method yg overrided jalan");
    }
    public void methodPemanggil (){
       System.out.println("Method pemanggil methodAsli jln");
       super.methodAsli();  // yg dipanggil milik kelas super
    }
    public static void main(String [] args) {
       SubKelas osk = new SubKelas();
       osk.methodAsli();
       osk.methodPemanggil();
    }
}
```

Output

```
Method yg overrided jalan
Method pemanggil methodAsli jln
Method milik KelasSuper jalan
Press any key to continue . . .
```

Menggunakan Method dan Constructor Kelas Super

- Constructor
  - super();
  - super(tipe parameter);

- Method (non static)
  - super.namaMethod();

Polymorphism

- artinya bersifat poly morphy (memiliki banyak bentuk)
- Method-method overloading dalam kelas yang sama
- Method-method overloading dalam kelas yang berbeda (kelas turunannya)

contoh 1

EkspresiWajah.java

```
class EkspresiWajah{
    public String respons() {
        return("Perhatikan ekspresi wajah saya");
    }
}
class Gembira extends EkspresiWajah{
    public String respons() {
        return("ha ha ha..");
    }
}
class Sedih extends EkspresiWajah{
    public String respons() {
        return("hik hik ngeee ngeee ngeee..");
    }
}
```

```
class Marah extends
EkspresiWajah{
    public String respons() {
        return("Hai kurang ajar..!");
    }
}
```

MainEkspresiWajah.java

```
class MainEkspresiWajah{
  public static void main(String args[]) {
    EkspresiWajah objEkspresi = new EkspresiWajah();
     Gembira objGembira = new Gembira();
     Sedih objSedih = new Sedih();
     Marah objMarah = new Marah();
     EkspresiWajah[] arrEkspresi = new EkspresiWajah[4];
     arrEkspresi[0] = objEkspresi;
     arrEkspresi[1] = objGembira;
     arrEkspresi[2] = objSedih;
     arrEkspresi[3] = objMarah;
     System.out.println("Ekspresi[0]: "+arrEkspresi[0].respons());
     System.out.println("Ekspresi[1]: "+arrEkspresi[1].respons());
     System.out.println("Ekspresi[2]: "+arrEkspresi[2].respons());
     System.out.println("Ekspresi[3]: "+arrEkspresi[3].respons());
    }
}
```

contoh 1

Output

```
Ekspresi[0] : Perhatikan ekspresi wajah saya
Ekspresi[1] : ha ha ha..
Ekspresi[2] : hik hik ngeee ngeee ngeee..
Ekspresi[3] : Hai kurang ajar..!
Press any key to continue . . .
```

contoh 2

Employee.java

```
public class Employee {
   private String name;
   private double salary;
   protected Employee(String n, double s) {
      name = n;
      salary = s;
   }
   protected String getDetails() {
      return "Name : "+name+ "\nSalary : "+salary;
   }
   public void cetak() {
      System.out.println("Coba di Employee");
   }
}
```

Manager.java

```
public class Manager extends Employee {
   private String department;
   public Manager(String nama, double salary, String dep) {
      super(nama, salary);
      department = dep;
   }
   public String getDepartment() {
      return department;
   }
   public String getDetails() {
      return super.getDetails()+"\nDepartment : "+getDepartment();
   }
   public void cetak() {
      System.out.println("Coba di Manager");
   }
}
```

View.java

```
public class View {
   public static void main(String[] args) {
      Employee e = new Employee("Ali",1200000);
      Employee em = new Manager("Ali",1200000,"Informatika");
      System.out.println("Data employee :\n"+e.getDetails());
      System.out.println("Data manager :\n"+em.getDetails());
      em.cetak();
   }
}
```

Output