

*Modul 4:  
Lebih Dalam lagi tentang Class,  
Objek dan Method*

Setelah mengikuti mata kuliah ini mahasiswa dapat memperdalam lagi bagaimana mendefinisikan class, membuat objek, mendefinisikan method

## Pengantar :

---

Dalam modul ini akan diuraikan beberapa topik bahasan yaitu :

1. Sekali lagi :Class dan objek
2. Mendefinisikan data dan method
3. Modifier untuk class dan method  
*:public, static, protected, final*

# 1. Mendalami Class

---

**Telah diuraikan dalam Modul 2 bahwa :**

- **Class** : suatu frame yang merupakan definisi yang memuat data dan metod pengolah data
- **Objek** : Bentuk konkrit dari class yang dengannya data dan metod dapat direalisasikan

**Dalam program class java dibagi menjadi :**

- Class yang didalamnya terdefinisi method **main()**. Class seperti ini bahwa jika beberapa class didefinisikan dalam file yang sama, maka nama file disimpan dalam nama sesuai nama class yang berisi main tersebut.
- Class yang didalamnya tidak terdapat metod **main()**. Class ini murni berisi encaptulation dari definisi data dimana didalamnya terdapat data dan method pengolah datanya.
- Pendefinisian class dan pembuatan objek juga dapat dilakukan dalam class yang sama sehingga bersifat rekursif.

## Definisi class Dog dan main yang rekursif

```
class Dog
{private String nama="Noname";
private int umur =0;
public Dog(){};
public Dog(String nm, int um){nama=nm; umur=um;}
public void bark(){System.out.println("Gug..gug..!");}
public void sayHello()
{System.out.println("Hello my name is "+nama+" I'm a "+umur+" years old");
}
public static void main (String[ ] arg)
{ Dog myDog= new Dog("Heli",3);
  Dog yourDog = new Dog("Pleki", 1);
  Dog hisDog=new Dog();
  myDog.bark();
  myDog.sayHello();
  System.out.println("Kenalkan anjingmu:");
  yourDog.sayHello();
  hisDog.sayHello();
} // akhir main
} //akhir class
```

Disini class Dog dalam main adalah rekursif

# Hasil program

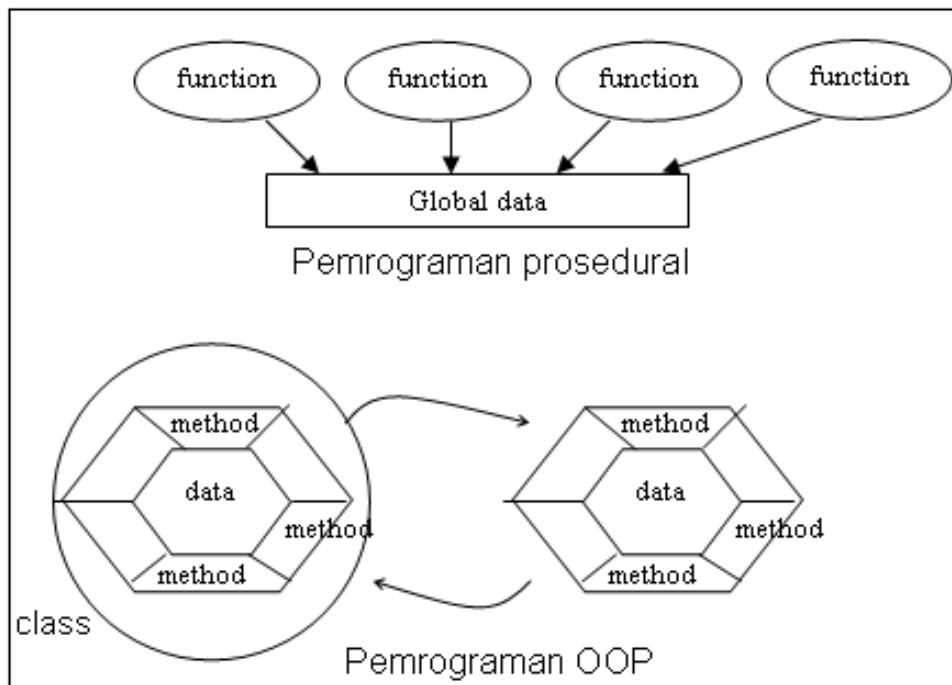
---



```
C:\j2\bin>javac Dog.java
C:\j2\bin>java Dog
Gug..gug..!
Hello my name is Heli Iffm a 3 years old
Kenalkan anjingmu:
Hello my name is Pleki Iffm a 1 years old
Hello my name is Noname Iffm a 0 years old
C:\j2\bin>
```

# Perbandingan OOP dan prosedural programming

- Berikut ini ditunjukkan perbedaan paradigma pemrograman OOP dengan pendekatan



Mendefinisikan method sesungguhnya mirip dengan mendefinisikan fungsi pada model perosedural

Metod dalam OOP merupakan bagian terpenting yang harus dikuasai seluk beluknya

## 2. Mendefinisikan Data dan Metod

---

### Mendefinisikan data

- Sesuai dengan ide *encapsulation*, data dalam suatu class sebaiknya didefinisikan secara **private**, agar ia lebih aman dari akses luar class. Jika tidak terpaksa pada data : berarti data hanya dapat diakses dari class yang sama
- Default dari definisi data adalah **public**, artinya jika kita tidak memberi keterangan apa-apa pada data maka data tersebut adalah **public**.
- Contoh definisi :  
`int a =9; int b=77;`  
definisi ini adalah berarti a dan b data yang dapat diakses dari luar class

**Perhatikan contoh berikut:**

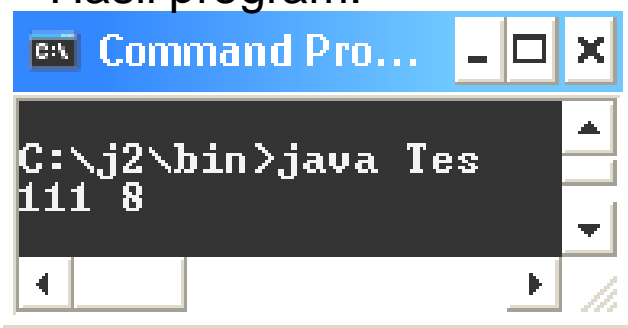
# Data default adalah public

- Program

```
public class Tes
{ public static void main(String args[])throws Exception
  {Data data=new Data();
   data.P=111;
   data.Tampil();
  }
}

class Data
{ int P=9;
  int Q=8;
  public void Tampil(){System.out.println(P+" "+Q);}
}
```

Hasil program:



```
Command Pro...
C:\j2\bin>java Tes
111 8
```

Perhatikan variabel P dan Q dalam definisi pada class Data

Tanpa keterangan P,Q adalah public

Dari class Tes, nilai P dan Q dapat diubah, perhatikan hasil program

Nilai P yang aslinya 9, dalam class Tes diubah menjadi 111



# Lindungi data default dengan modifier private

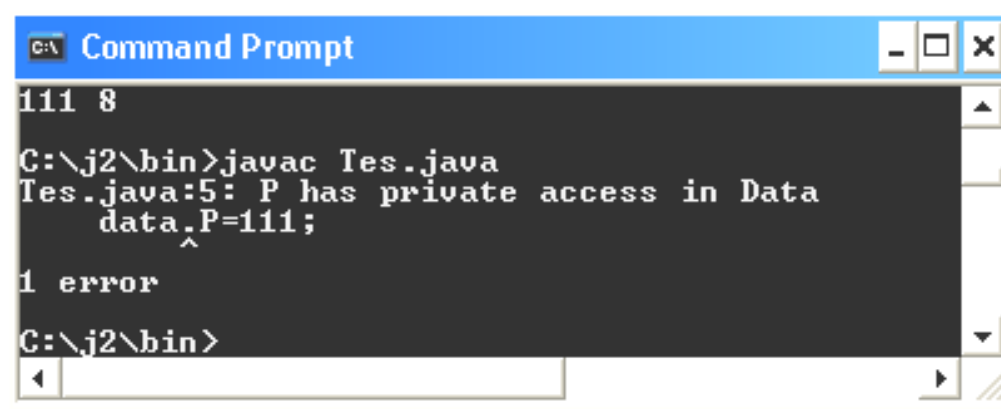
---

Jika statemen definisi data untuk P dan Q dibuat private :

```
private int P=9;
```

```
private int Q=8;
```

Maka pada saat kompilasi akan gagal untuk melakukan akses data private dari luar class, seperti pesan kesalahan kompilasi berikut.



```
C:\ Command Prompt
111 8
C:\j2\bin>javac Tes.java
Tes.java:5: P has private access in Data
    data.P=111;
        ^
1 error
C:\j2\bin>
```

Gunakan metode `set(.)` untuk ubah data `private` dan metode `get()` untuk ambil data `private` dari luar class

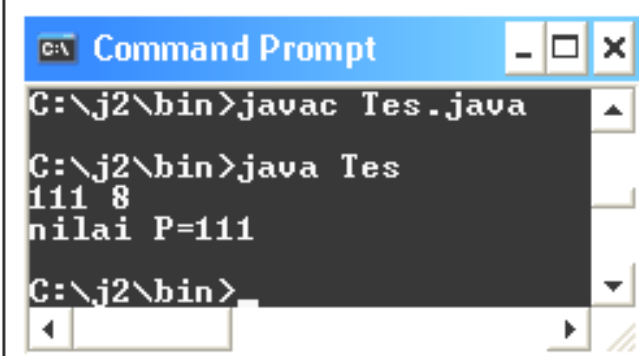
---

Data **private** masih diberi kemungkinan perubahan dengan mendefinisikan metode `set()` untuk menggantinya dan metode `get()` untuk mengambilnya.

**Contoh :**

untuk merubah data `P` yang **private** dalam class `Data` tambahkan definisi `setP()` dan `getP()` sebagai berikut :

```
public class Tes
{ public static void main(String args[])throws Exception
  {Data data=new Data();
   data.setP(111);
   data.Tampil();
   System.out.println("nilai P="+data.getP());
  }
}
class Data
{ private int P=9;
  private int Q=8;
  public void setP(int p){P=p;}
  public int getP(){return P;}
  public void Tampil(){System.out.println(P+" "+Q);}
}
```



```
C:\j2\bin>javac Tes.java
C:\j2\bin>java Tes
111 8
nilai P=111
C:\j2\bin>
```

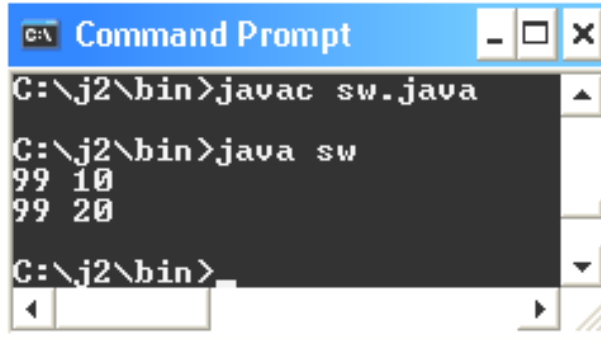
# Data static

Data **static** adalah data yang didefinisikan untk dimiliki oleh class. Jika objek berasal dari class yang sama maka data static akan bernilai sama untuk seluruh objek dalam class tersebut.

```
import java.io.*;
public class sw
{ public static void main(String args[])
  {sw1 a=new sw1(); sw1 b=new sw1();
   a.P=111; a.Q=10; b.P=99; b.Q=20;
   a.Tampil();b.Tampil();
  }
}

class sw1
{ static int P=9;
  int Q=8;
  void Tampil(){System.out.println(P+" "+Q);}
}
```

Hasil program



```
C:\j2\bin>javac sw.java
C:\j2\bin>java sw
99 10
99 20
C:\j2\bin>
```

Jika nilai P diubah pada suatu objek, maka nilai P pada objek lain akan ikut berubah. Pada objek b nilai P diubah menjadi 99, ternyata pada objek a nilai P ikut berubah menjadi 99

# Contoh efek kata : static

```
class Mahasiswa
{ String Nama;
  int Umur;

  private void SetData(String n, int u)
  { Nama=n; Umur=u;};

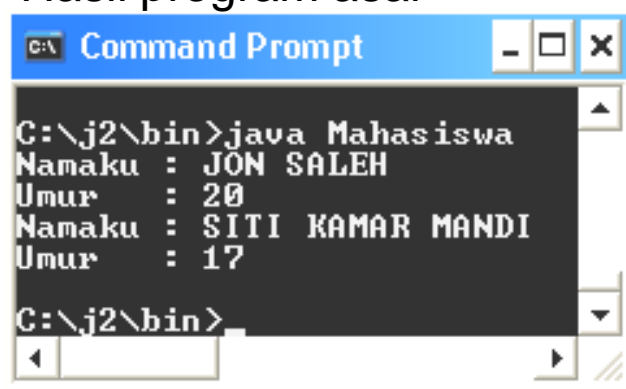
  public void CetakInfo()
  { System.out.println("Namaku : "+Nama);
    System.out.println("Umur   : "+Umur);
  }
  public static void main(String orgs[])
  { Mahasiswa Mhs1=new Mahasiswa();
    Mahasiswa Mhs2=new Mahasiswa();

    Mhs1.SetData("JON SALEH",20);
    Mhs2.SetData("SITI KAMAR MANDI",17);

    Mhs1.CetakInfo();
    Mhs2.CetakInfo();
  }
}
```

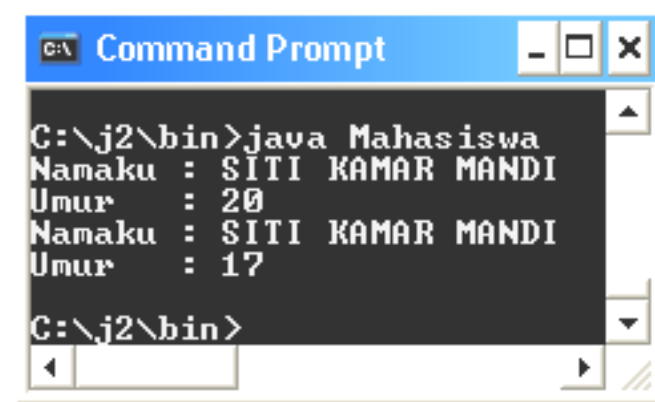
Jika dalam deklarasi nama diganti dengan **static String Nama**

Hasil program asal



```
C:\j2\bin>java Mahasiswa
Namaku : JON SALEH
Umur   : 20
Namaku : SITI KAMAR MANDI
Umur   : 17
C:\j2\bin>
```

static String nama;



```
C:\j2\bin>java Mahasiswa
Namaku : SITI KAMAR MANDI
Umur   : 20
Namaku : SITI KAMAR MANDI
Umur   : 17
C:\j2\bin>
```

# Metod :

## Default Definisi metod: public

---

**Definisi metod tanpa modifier adalah public**

**Contoh :**

```
class Data
{   int P=9;
    int Q=8;
    void Tampil(){System.out.println(P+" "+Q);}
}
public class Tes
{   public static void main(String args[])
    {Data data=new Data();
     data.P=111;
     data.Tampil();
    }
}
```

Tanpa modifier  
berarti public

**Hasil program :**



```
C:\>j2\bin>javac Tes.java
C:\>j2\bin>java Tes
111 8
```

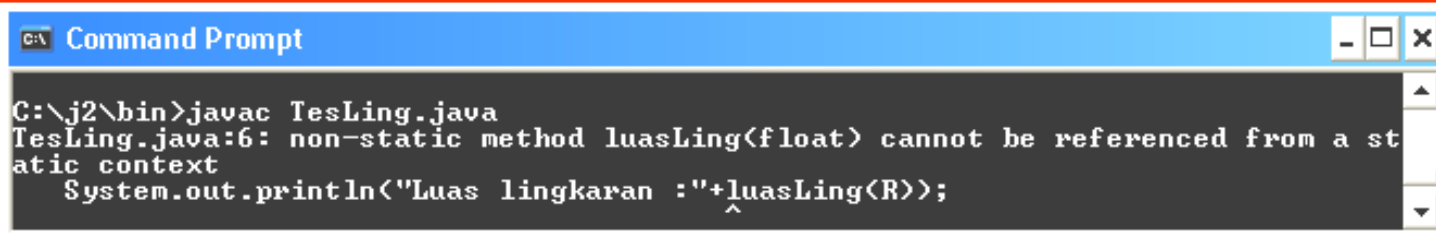
# Metod static

**Metod static adalah metod yang diases oleh metod static yang lain dari dalam class yang sama**

**Contoh : definisi fungsi luasLing(float R)**

```
public class TesLing
{ public static void main(String args[])
  {float R=1.0f; float T=10;
   System.out.println("Jari-jari :"+R);
   System.out.println("Luas lingkaran :"+luasLing(R));
  }
  private float luasLing(float R){return (22f/7)*R*R ;}
}
```

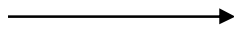
Dalam definisi **luasLing(float R)** kata **static** tidak ditemukan, tetapi ada kata **private**. Saat dikompile ternyata salah.. Karena metod main yang memanggilnya adalah suatu metod static



```
C:\> Command Prompt
C:\j2\bin>javac TesLing.java
TesLing.java:6: non-static method luasLing(float) cannot be referenced from a static context
    System.out.println("Luas lingkaran :"+luasLing(R));
                                   ^
```

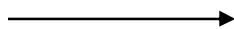
# Setelah kata static dipasang

Static  
dipasang



```
public class TesLing
{ public static void main(String args[])
  {float R=1.0f; float T=10;
   System.out.println("Jari-jari :"+R);
   System.out.println("Luas lingkaran :"+luasLing(R));
  }
  private static float luasLing(float R){return (22f/7)*R*R ;}
}
```

Hasil  
program

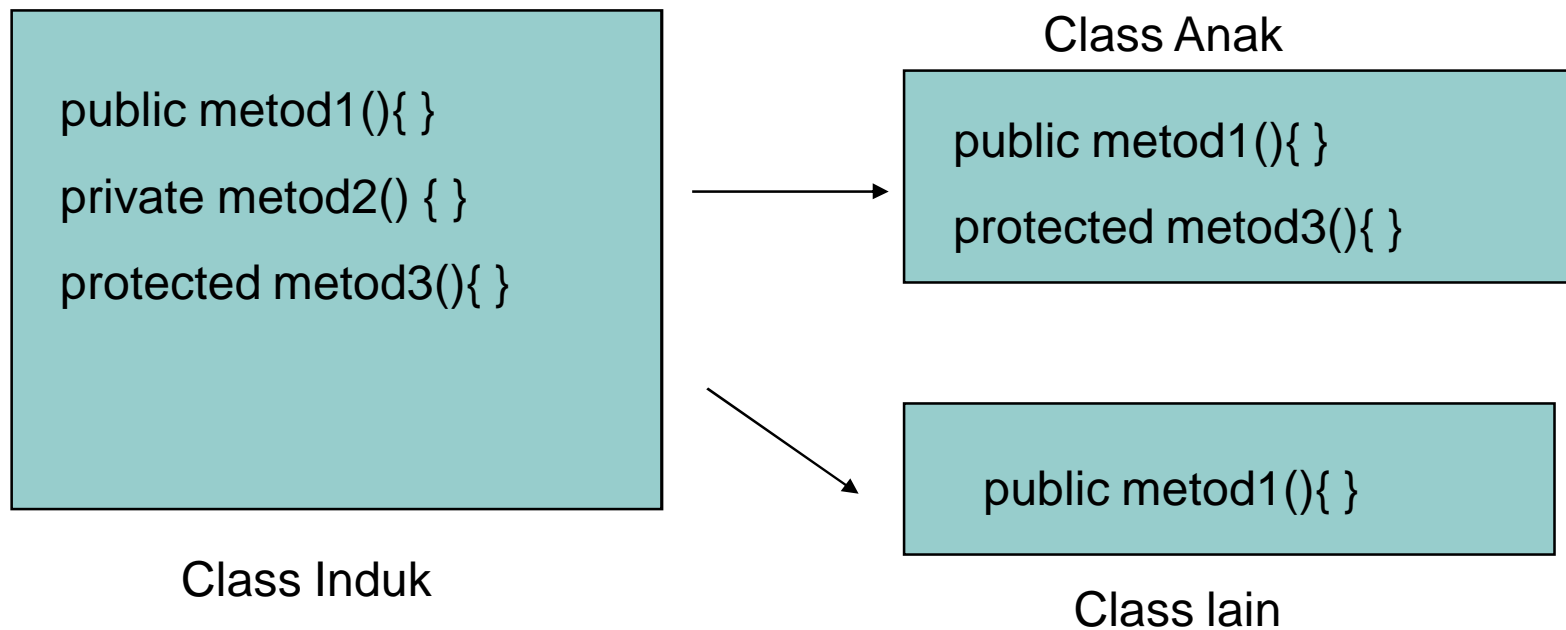


```
C:\> javac TesLing.java
C:\> java TesLing
Jari-jari :1.0
Luas lingkaran :3.142857
```

Kata private pada metod menunjukkan bahwa metod tersebut diases oleh metod dalam class yang sama

# Metod protected

- **protected**: menunjukkan bahwa metod tersebut hanya dapat diwarisi oleh class turunannya.
- Dalam pembahasan akan diuraikan dalam bagian penurunan (**inheritance**) class

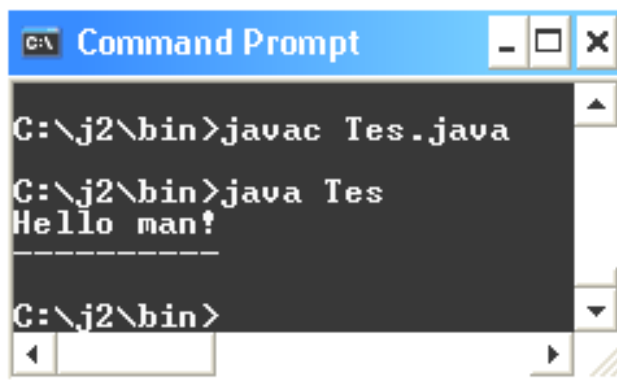




## 3. Class static

- Jika suatu class didefinisikan tidak untuk dibuat objeknya, tetapi untuk dimanfaatkan metod-metodnya maka class tersebut adalah **class static**.
- Dalam java contoh class static adalah class Math, dimana metod-metod seperti random(), sqrt() dan lain-lain diambil dari class ini, tetapi class ini tidak pernah dibuat objeknya.
- Contoh : **class Data** berikut adalah class static dimana ia hanya digunakan metodnya, tidak dibuat objeknya

```
public class Tes
{ public static void main(String args[])
  { Data.Tampil("Hello man!");
    Data.Garis();
  }
}
class Data
{ public static void Tampil(String s)
  {System.out.println(s);}
  public static void Garis()
  {System.out.println("-----");}
}
```



```
C:\> Command Prompt
C:\j2\bin>javac Tes.java
C:\j2\bin>java Tes
Hello man!
-----
C:\j2\bin>
```

# Class final

---

- Jika suatu class didefinisikan secara final maka class tersebut tidak dapat dibuat turunannya
- Contoh class-class final dalam java adalah :
- class Math
- class String
- class Integer, Long, Double, Float

# Rangkuman

---

1. Modifier **private** untuk data dan metod berarti hanya dapat diakses oleh metod dari class yang sama
2. Modifier **public** untuk data dan metod berarti dapat diakses dari luar class
3. Modifier **public** untuk class berarti berisi main program dan harus disimpan dengan nama yang sama dengan nama class tersebut
4. Modifier **protected** untuk data dan metod berarti dapat diwarisi oleh class turunannya
5. Modifier **static** untuk data berarti bernilai sama untuk semua objek dari class yang sama
6. Modifier **static** untuk metod berarti dapat diakses dari metod static yang sama dalam class
7. Modifier **final** untuk class berarti tak dapat dibuat turunannya

# Latihan

---

1. Buatlah definisi class MahasiswaTeknik dan MahasiswaSastra.  
Masing-masing class berisi data:  
JUR (String, static)  
NIP(int, private)  
Nama(String, private)  
Masing-masing class berisi metod :  
setNamaNIP(String n ,int m) : untuk mengisi NIP dan Nama  
setJUR(String s) untuk mengatur nama Jurusan  
tampilkan() untuk menampilkan data NIP NAMA dan JURUSAN
2. Buatlah class TesMahasiswa yang berisi main program
3. Buat objek MahasiswaTeknik 3 orang dan MahasiswaSastra 2 orang

# Latihan ...

---

4. Buatlah tampilan daftar mahasiswa dengan tampilan :

```
-----  
No  NIP  NAMA          JURUSAN  
-----  
  
-----
```