



Pemrograman Berorientasi Objek

Oleh:
Mike Yuliana
PENS-ITS

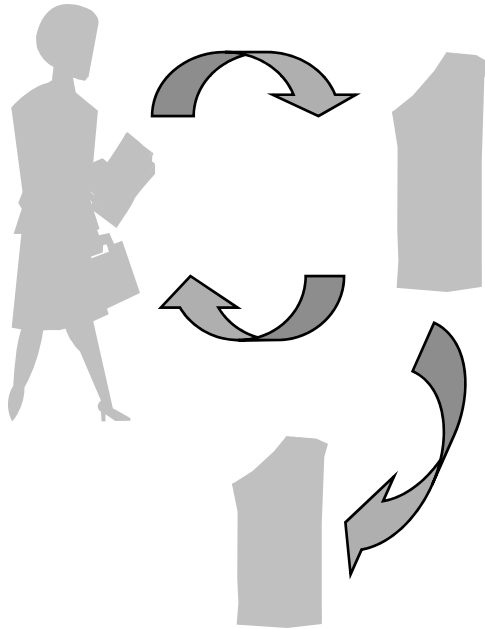


Tujuan

- Konsep pemrograman berorientasi objek
- Menciptakan kelas
- Membuat objek dari suatu kelas
- Mengakses variabel dan method dari suatu kelas
- Kata Kunci this
- Penentu Akses: public dan private
- Kata Kunci static

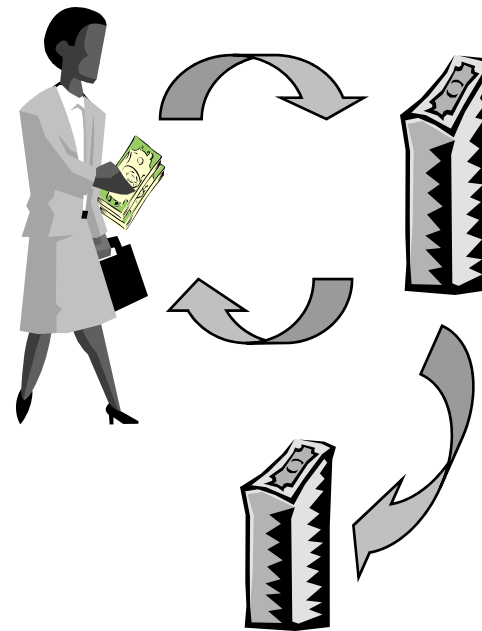
Konsep pemrograman berorientasi Objek

- Procedural



mengambil, menabung, mentransfer

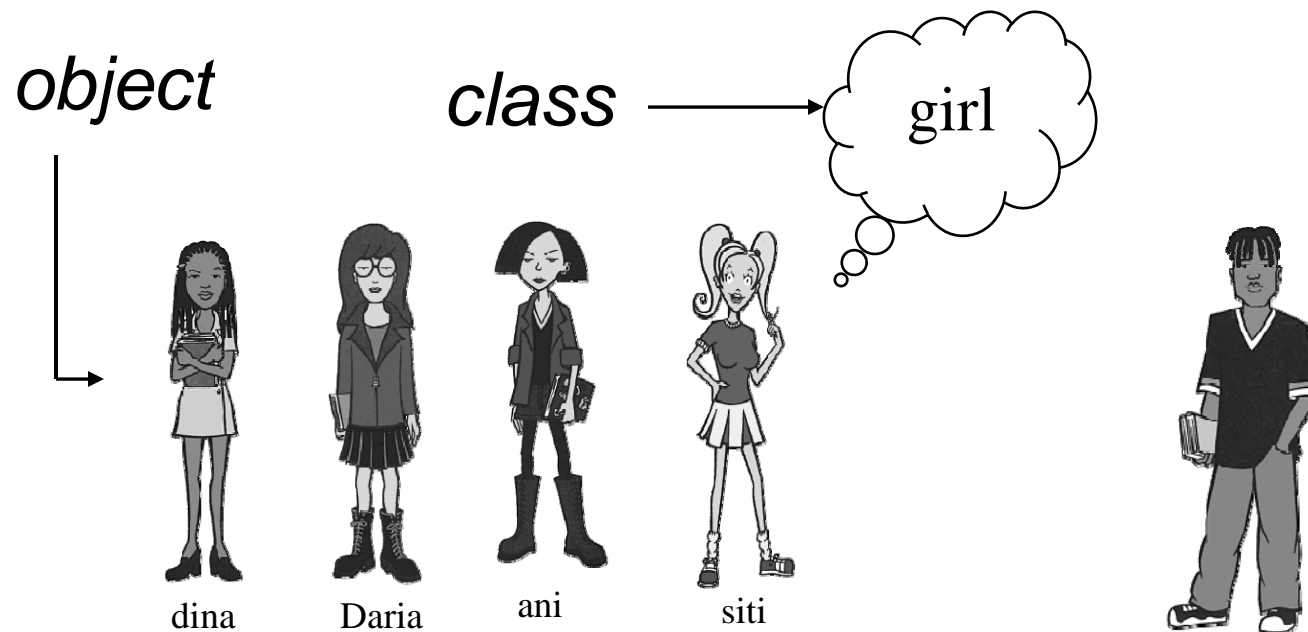
- Object Oriented

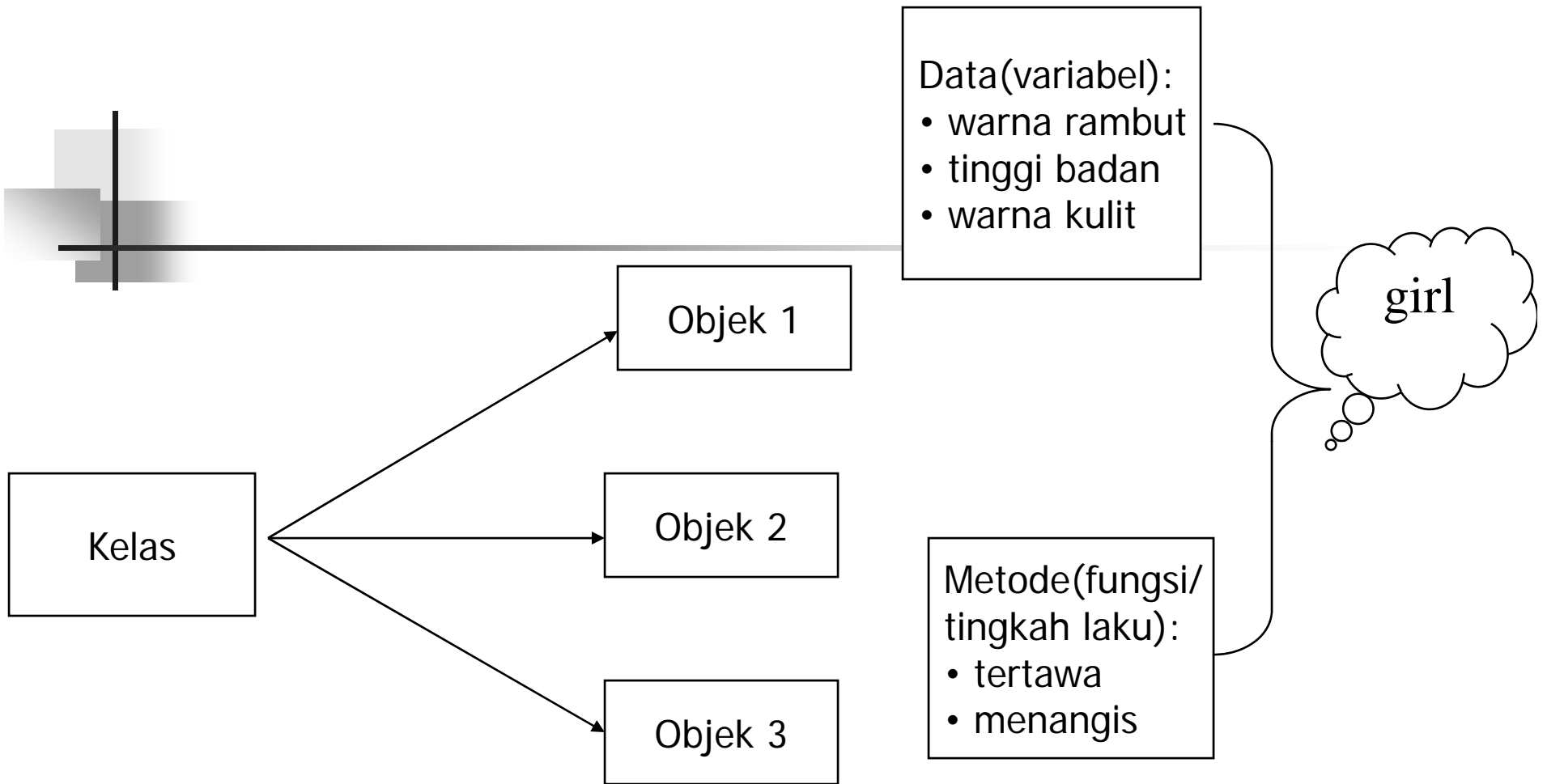


pelanggan, uang, rekening

Objek dan kelas

- Kelas adalah pemodelan dari objek yang berisi informasi tentang karakteristik(data) dan tingkah laku yang dimiliki oleh objek tersebut(metode), sedangkan objek merupakan perwujudan dari suatu kelas .





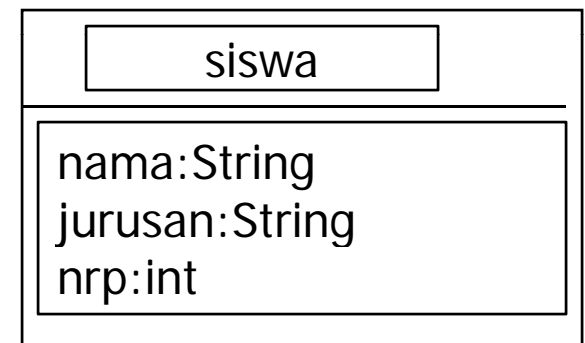
Menciptakan Kelas

- Kelas pada java didefinisikan dengan menggunakan kata kunci class
- Bentuk umum:

```
class namakelas
{
    class body
}
```

- Contoh:

```
class siswa
{
    String nama;
    String jurusan;
    int nrp;
}
```



UML class diagram



Membuat Objek dari Suatu Kelas

- Dibutuhkan operator new untuk membuat objek dari suatu kelas
- Bentuk umum:

```
namakelas variabelreferensiobjek=new namakelas();
```

- Contoh:

```
siswa a=new siswa();
```



Ungkapan diatas merupakan bentuk singkat dari proses berikut:

- Mendeklarasikan variabel referensi objek
siswa a;
- Menciptakan objek dari kelas, dengan operator new
new siswa();
- Menugaskan(meng-assign) variabel kepada objek yang telah dibuat
a=new siswa();

Mengakses Variabel dari suatu kelas

Program siswaku.java:

```
class siswa
{
    String nama;
    String jurusan;
    int nrp;
}
public class siswaku
{
    siswa a=new siswa();
    a.nama="cita";
    a.jurusan="telkom";
    a.nrp=7206;
    System.out.println("nama" + a.nama + "jurusan"
    + a.jurusan+"nrp"+a.nrp);
}
```

variabel

variabelreferensiobjek.nama variabel

Mengakses Variabel dan Metode dari suatu kelas

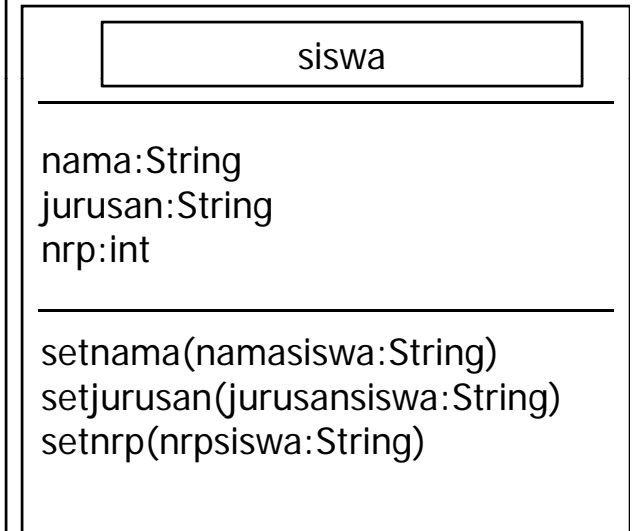
Contoh:siswalg.java

```
class siswa
{
    String nama;
    String jurusan;
    int nrp;

    void setnama(String namasiswa)
    {
        nama=namasiswa;
    }
    void setjurusan(String jurusansiswa)
    {
        jurusan=jurusansiswa;
    }
    void setnrp(String nrpsiswa)
    {
        nrp=nrpsiswa;
    }
}
public class siswalg
{
    →
}
```

metode

Variabelreferensiobjek.nama metode



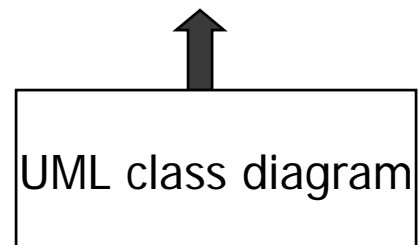
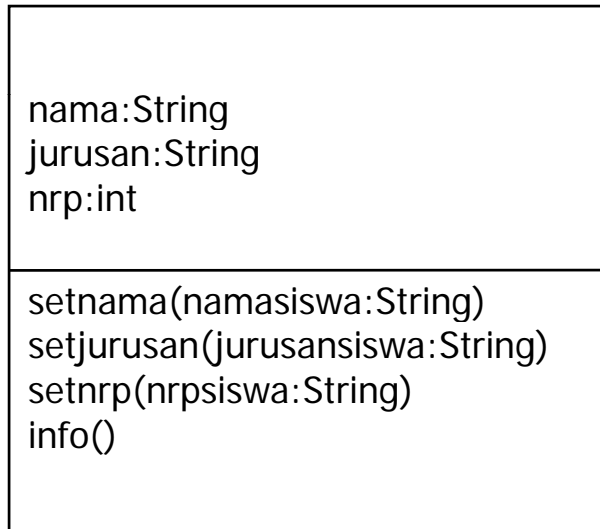
UML class diagram

Apa yang harus ditambahkan
Agar muncul tampilan:
Cita adalah namaku telkom adalah
Jurusanku dan nrpkku adalah 7206

Soal 1

- Buatlah program dari UML class diagram dibawah ini sehingga didapat tampilan sbb:

```
nama:cita
jurusan:telkom
nrp:7206
nama:raka
jurusan:it
nrp:7306
```



Soal 2

- Buatlah program dari UML class diagram dibawah ini sehingga didapat tampilan sbb:

nama:cita

jurusan:telkom

nrp:7206

nama:raka

jurusan:it

nrp:7306

```
nama:String  
jurusan:String  
nrp:int
```

```
setnama(namasiswa:String)  
setjurusan(jurusasiswa:String)  
setnrp(nrpsiswa:String)  
getnama():String  
getjurusan():String  
getnrp():int
```

UML class diagram



Kata Kunci(Keyword) This

Kata kunci this dipergunakan pada pembuatan kelas dan digunakan untuk menyatakan objek sekarang

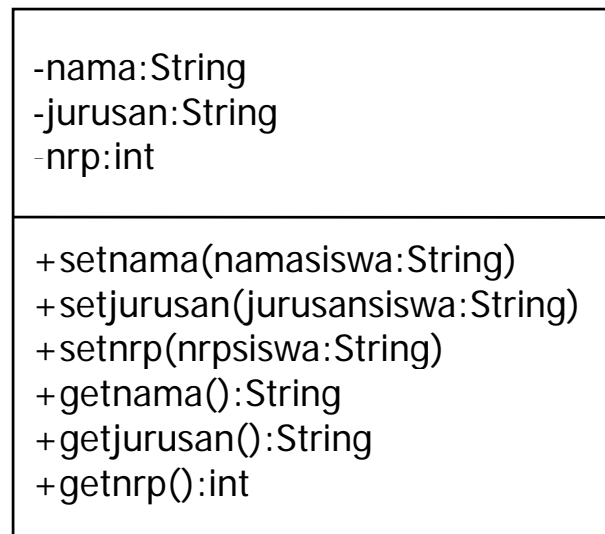
```
class siswa
{
    String nama;
    String jurusan;
    int nrp;
    void setnama(String nama)
    {
        nama=nama;
    }
    void setjurusan(String jurusan)
    {
        jurusan=jurusan;
    }
    void setnrp(String nrp)
    {
        nrp=nrp;
    }
}
```

Akan menimbulkan
kesalahan!!!

Penentu Akses: public dan private

public: pengaksesan suatu variabel atau metode dapat dilakukan diluar kelas

Private: pengaksesan suatu variabel atau metode hanya dapat dilakukan di dalam kelas, tidak bisa diakses dari luar kelas



UML class diagram

Kata Kunci(Keyword) Static

Bentuk umum: nama kelas.nama metode

bukan

variabelreferensiobjek.nama metode

Contoh:

```
class matematika
{
    static public double kuadrat(double nilai)
    {
        return nilai*nilai;
    }
}
```

```
public class coba
```

```
{
    public static void main(String args[])
    {
        double bilangan=matematika.kuadrat(25.0);
        System.out.println(bilangan);
    }
}
```

nama kelas.nama metode

contoh:

```
public class coba4{
static int pencacah=0;
int nilai;

coba4(int nilai){
this.nilai=nilai;
this.pencacah++;
}

public void info(){
System.out.println(this.nilai);
System.out.println(this.pencacah);
}
}
```

```
class coba{
public static void main(String args[]){
coba4 a=new coba4(4);
a.info();

coba4 b=new coba4(5);
b.info();

coba4 c=new coba4(6);
c.info();
}
}
```

Hasil:

4
1
5
2
6
3

Contoh lain:

```
public class coba5{
static int x=10,y=20;
}

class coba{
public static void main(String args[]){
coba5 a=new coba5( );
coba5 b=new coba5( );

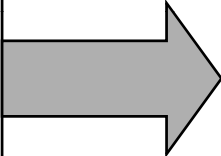
System.out.println(a.x);
System.out.println(a.y);

b.x=25;
b.y=55;

System.out.println(a.x);
System.out.println(a.y);

}
}
```

Hasil:



```
10
20
25
55
```