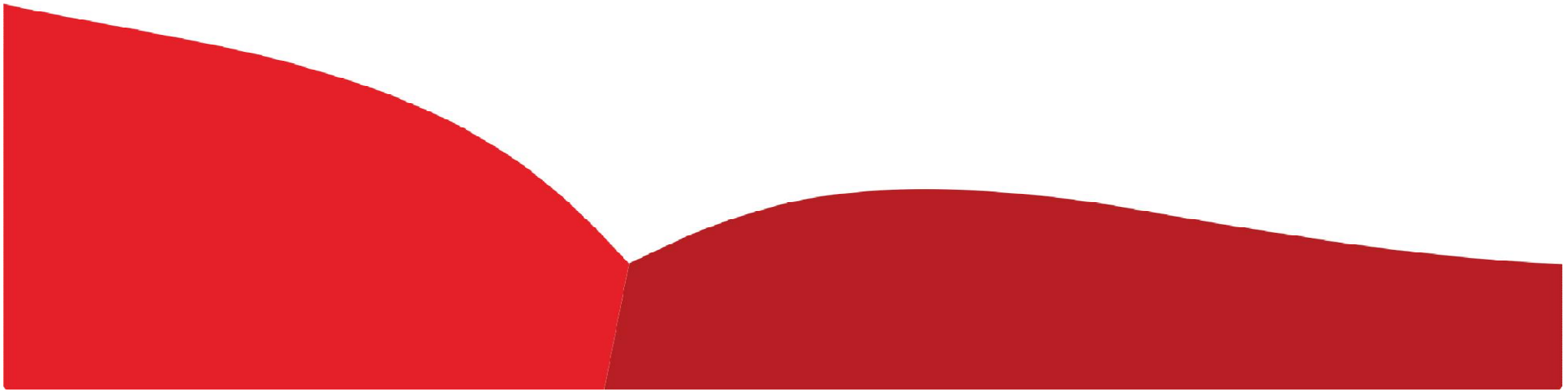
Relasi Antar Kelas

Oleh : Agus Priyanto, M.Kom



Tujuan Perkuliahan

- Mengerti tentang relasi kelas
- Mengerti dan mampu mewujudkan berbagai jenis relasi kelas C++ dan Java

Outline Materi

1. Pengantar
2. Pewarisan
3. Agregasi
4. Asosiasi
5. Dependensi

Pengantar

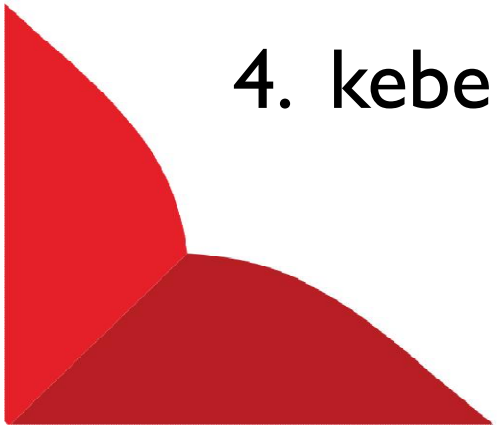
- Pemrograman berorientasi objek mengambil realita dari kehidupan sehari-hari
- Antara satu objek dengan objek yang lain sering terjadi relasi
  - Komunikasi antara satu objek dengan objek terjadi

- Sejauh ini kita lebih banyak membicarakan tentang kelas tanpa memberi penekanan pada relasi yang mungkin terjadi antar kelas.

- Pewarisan …

- Ada beberapa relasi yang mungkin terjadi dalam antara satu kelas dengan kelas yang lain:

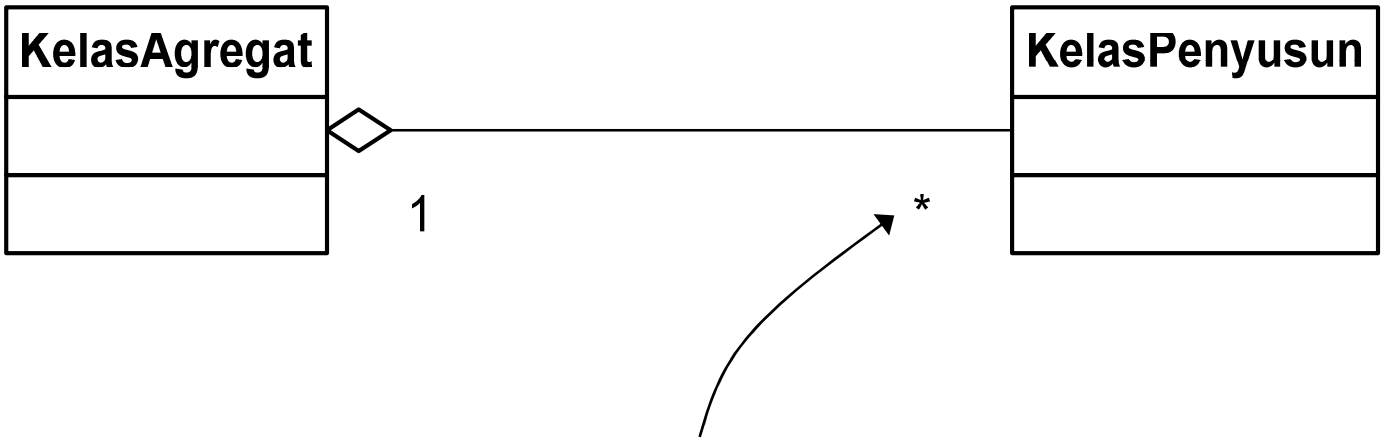
1. Pewarisan (*inheritance*)
2. Agregasi (*Aggregation*)
3. Asosiasi (*association*)
4. kebergantungan (*dependency*)



Agregasi

- Relasi antara dua objek dengan mengatakan bahwa satu objek memiliki atau mengandung atau berisi objek yang lain
  - mobil memiliki mesin
  - rumah memiliki dapur
  - fakultas memiliki jurusan

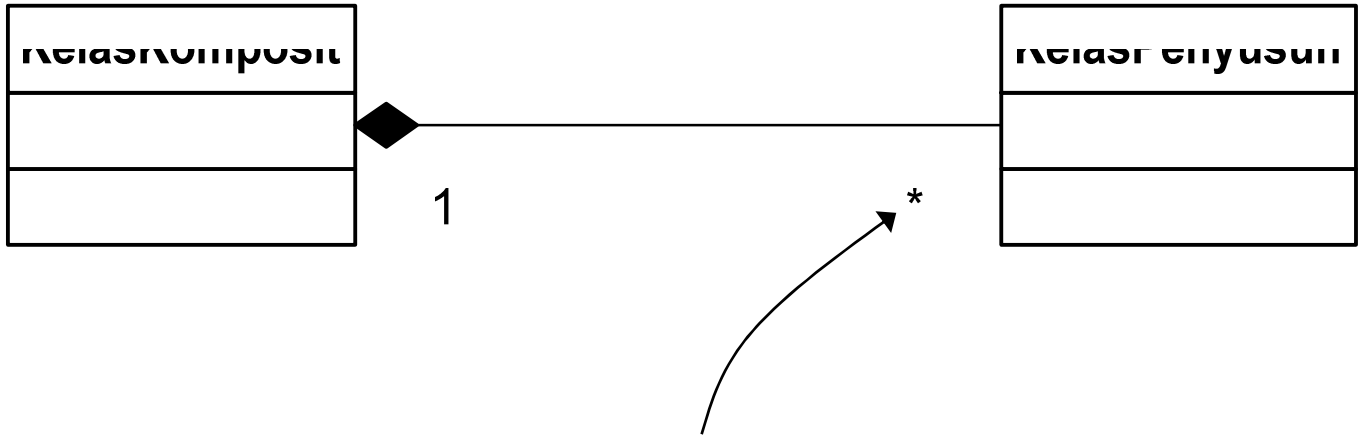
- Diagram UML


KelasAgregat
KelasPenyusun
1
*
Beberapa / tidak tentu jumlahnya

Komposisi

- Varian dari agregrasi
- Strong aggregation
- Objek dari kelas penyusun hanya ada selama objek kelas komposit ada

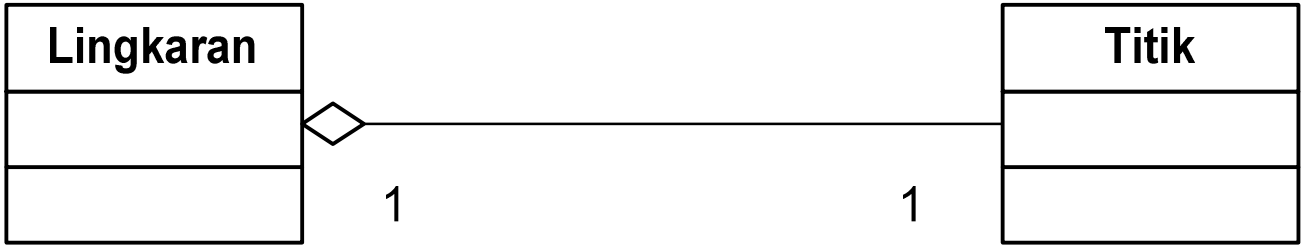
- Diagram UML



Beberapa / tidak tentu jumlahnya

Agregasi – Contoh 1

- Lingkaran dengan titik pusat



Agregasi – Contoh 2

class Titik

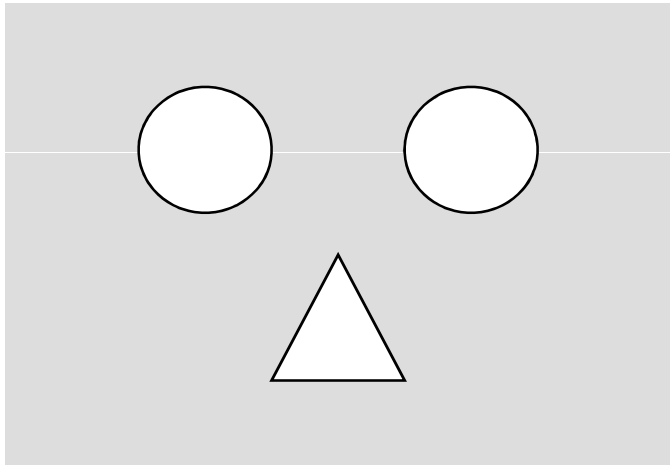
```
{
        // indeks koordinat dari sebuah titik
        float sbX, sbY;
        /**
         * Membuat objek titik dengan posisi di pusat koordinat
         * (0,0)
         */
        public Titik()
        {
                sbX = sbY = 0.0f;
        }
        /**
         * Membuat objek titik dengan posisi di tertentu
         * (sbX,sbY)
         */
        public Titik(float x, float y)
        {
                sbX = x;
                sbY = y;
        }
// kode yang lain dihilangkan
}
```

class Lingkaran

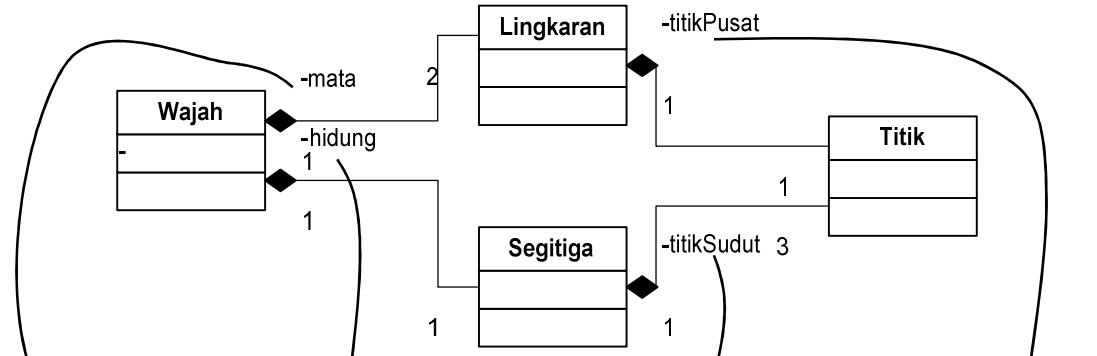
```
class Lingkaran
{
        // atribut dari sebuah lingkaran
        float jejari;
        Titik titikPusat; // menunjukkan relasi agregasi
                          //antara Titik dan Lingkaran
                        // 1 Lingkaran memiliki 1 titik pusat

        // bagian yang lain tidak ditunjukkan
}
```

Agregasi – Contoh 2

- Bagaimana dengan ini … ?



Agregasi – Contoh 2




```
class Wajah
{
    Lingkaran mata[ ];
    Segitiga hidung;
    .....
}
```

```
class Segitiga
{
    Titik titikSudut[ ];
    .....
}
```

```
class Lingkaran
{
    Titik titikPusat;
    .....
}
```

Asosiasi

- Bagaimana relasi yang terjadi antara objek dosen dengan mata kuliah ?

- Bagaimana relasi yang terjadi antara objek mahasiswa dengan mata kuliah ?

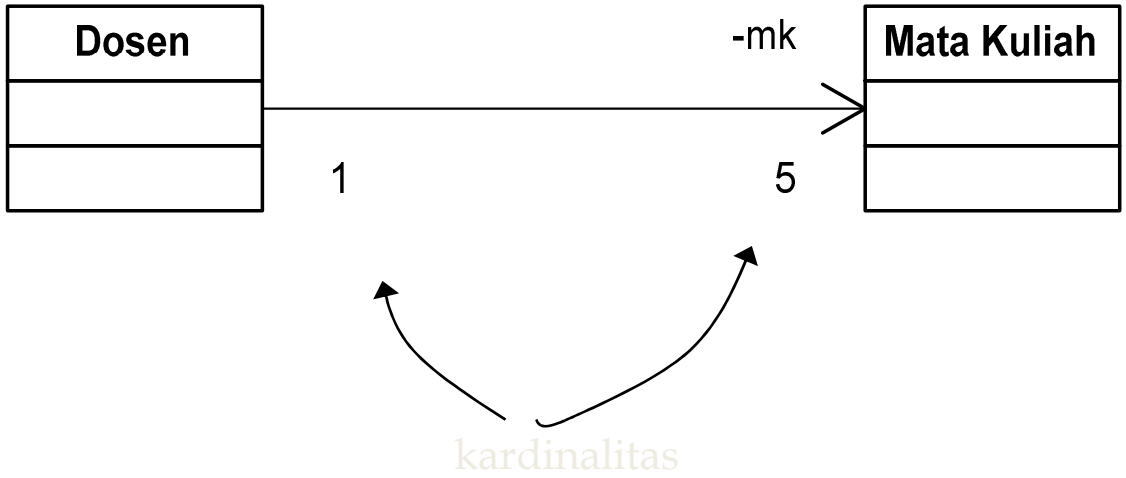
- Dalam Relasi perlu diperhatikan Kardinalitas
  - berapa objek yang terlibat dari masing-masing kelas yang terlibat.
  - apakah relasi tersebut bersifat wajib (*mandatory*) atau opsional.

- menyatakan suatu hubungan struktural antar objek yang menggambarkan objek dari suatu kelas dihubungkan ke objek dari kelas lain
- menunjukkan variabel dalam suatu kelas yang menyimpan rujukan bertipe kelas lain

Asosiasi

- menyatakan suatu hubungan struktural antar objek. yang menggambarkan objek dari suatu kelas dihubungkan ke objek dari kelas lain
- menunjukkan variabel dalam suatu kelas yang menyimpan rujukan bertipe kelas lain

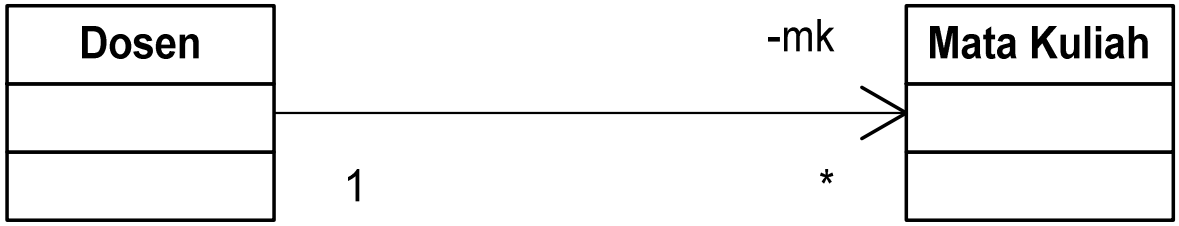
- Diagram UML



- Dalam implementasi, secara sintaks tidak memiliki perbedaan dengan implementasi agregasi, kecuali asosiasi bersifat dua arah

- Lihat Asosiasi.java

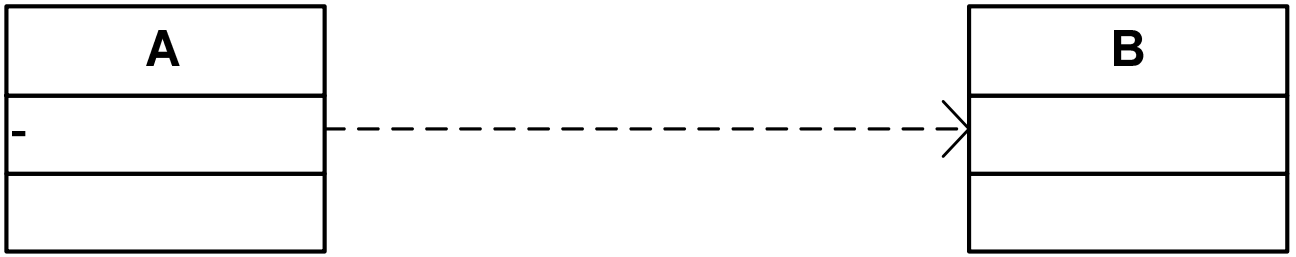
- Bagaimana dengan relasi berikut ini …



Dependensi

- Suatu kelas A bergantung pada kelas B → jika kelas B mengalami perubahan maka kelas A akan terkena dampak perubahan tersebut

- Diagram UML



- Perwujudan relasi ini dapat dilakukan dalam 3 bentuk.

1. Penggunaan kelas B sebagai parameter pada fungsi di kelas A

```
class B { ... }
class A
{
    void fungsiA(B varB) { ..... }
}
```

2. Penggunaan kelas B sebagai nilai balikan pada fungsi di kelas A

```
class B { ... }
class A
{
      B fungsiA(....) { ..... }
}
```

3. Penggunaan kelas B sebagai variabel lokal pada fungsi di kelas A

```
class B { ... }
class A
{
      void fungsiA(....)
{
      B varLokal;
 }
}
```

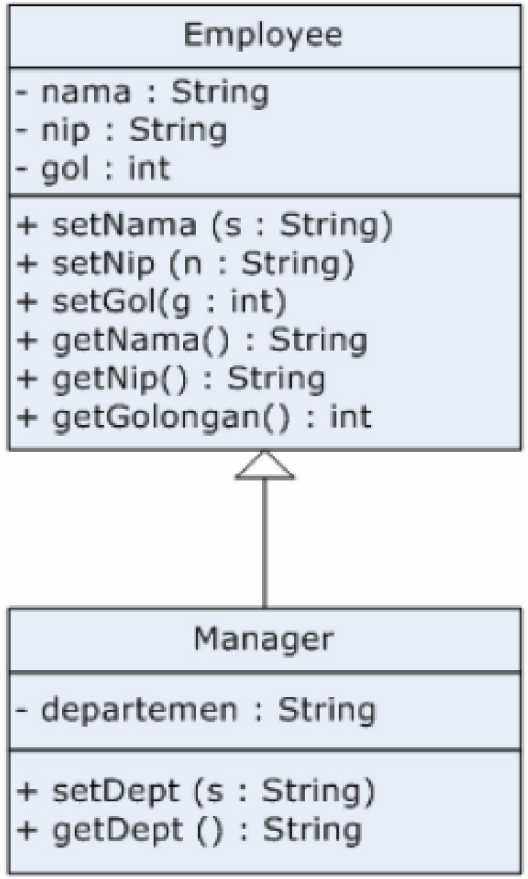
Penutup

- Mengerti Relasi dan mengimplementasikannya dalam bahasa Pemrograman

Tugas Terstruktur 1

Keterangan diagram kelas di atas:

- Golongan seorang employee berkisar dari 1 s.d. 7
- Buatlah kode Java untuk diagram kelas di atas!




Backspace
Thank You
Page Up
Home
Page Down
Shift
Insert