# CLASS RELATION (RELASI KELAS)

PEMROGRAMAN BERBASIS OBJEK – PERTEMUAN 4

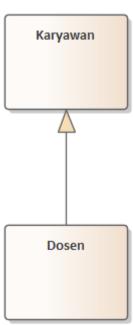


### JENIS RELASI

- Ada 4 jenis relasi dasar antar class:
  - I. Inheritance
  - 2. Association
  - 3. Aggregation
  - 4. Composition

#### **INHERITANCE**

- Menggambarkan relasi antara class yang lebih umum (parent/superclass) dan class yang lebih khusus (child/subclass)
- "is-a" relationship
- Contoh:
  - Dosen adalah seorang Karyawan
  - Kucing adalah seekor Binatang
  - Gedung adalah sebuah Bangunan
  - Mobil merupakan sebuah Kendaraan

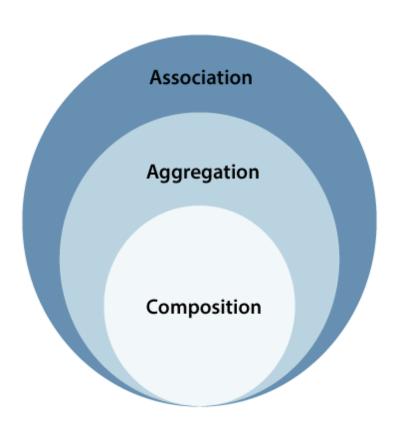


### **ASSOCIATION**

- Asosiasi merupakan relasi antar objek yang saling berkaitan dalam suatu sistem
- "has-a" relationship
- Contoh:
  - Dosen dan MataKuliah
  - Mahasiswa dan MataKuliah

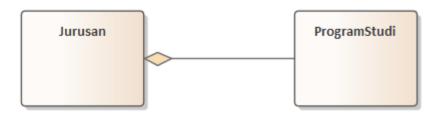


## **ASSOCIATION**



#### **AGGREGATION**

- Aggregation merupakan bentuk khusus dari Association dimana terdapat ownership antara satu class terhadap class lain, namun masing-masing objek memiliki life-cycle nya sendiri
- Unidirectional association/one-way relationship
- Ownership artinya ada kepemilikan antara kedua class tersebut. Contoh:
  - Jurusan terdiri dari beberapa Program Studi
  - Program Studi merupakan bagian dari Jurusan
- Jurusan dan Program Studi memiliki life-cyle nya masing-masing, artinya jika
   Jurusan dihapus, Program Studi tidak secara otomatis terhapus juga.



#### COMPOSITION

- Composition merupakan bentuk khusus dari Aggregation dimana terdapat ownership antarkelas tetapi child object tidak memiliki life-cycle sendiri tanpa parent object
- Tidak memiliki life cycle sendiri artinya, child object tidak dapat dibuat tanpa ada parent object; child object juga akan terhapus jika parent object dihapus.
- Contoh:
  - Gedung terdiri dari Ruangan
  - Karyawan memiliki Tanggungan
  - E-mail *memiliki* Attachment

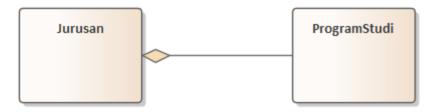


### JURUSAN & PROGRAM STUDI

• Jika jurusan dihapus, maka otomatis program studi dihapus  $\rightarrow$  composition



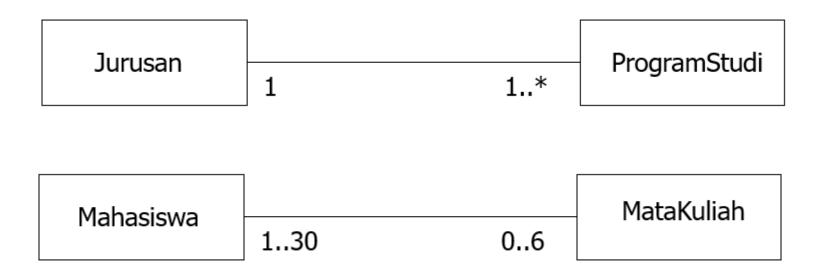
■ Jika jurusan dihapus, program studi bisa berpindah ke jurusan lain  $\rightarrow$  aggregation



### MULTIPLICITY NOTATION

Contoh	Keterangan
I	Tepat satu
01	Nol atau satu
0*	Nol atau lebih
1*	Satu atau lebih
35	3 hingga 5

### MULTIPLICITY NOTATION



Pasien				Konsultasi	
- noRekamMedis: String - nama: String - riwayatKonsultasi: ArrayList <konsultasi></konsultasi>		- d	anggal: Lo lokter: Peg erawat: P	jawai	
+ Pasien(noRekamMedis:String, nama:String) + getNoRekamMedis (): String + setNoRekamMedis (nip: String): void + getNama(): String + setNama(nama: String): void + getInfo(): String + tambahKonsultasi(tanggal:LocalDate, dokter:Pegawai, perawat:Pegawai)	1 0*	+ 9 + 8 + 9 + 8 + 9	+ getTanggal(): LocalDate + setTanggal(tanggal: LocalDate): void + getDokter(): Pegawai + setDokter(dokter: Pegawai): void + getPerawat(): Pegawai + setPerawat(perawat: Pegawai): void + getInfo(): String		
			0*		0*
		mе	merawat memeril		
			1		1
			Pegawai		
			- nip: String - nama: String		
		+ Pegawai(nip:String, nama:String) + getNip(): String + setNip(nip: String): void + getNama(): String + setNama(nama: String): void + getInfo(): String			

### **LATIHAN**

- Buat sebuah studi kasus dengan minimal 3 class yang saling berasosiasi.
- Tentukan atribut dan method pada masing-masing class
- Gambarkan class diagramnya.