

Relationship Between Classes

Chapter III

Content

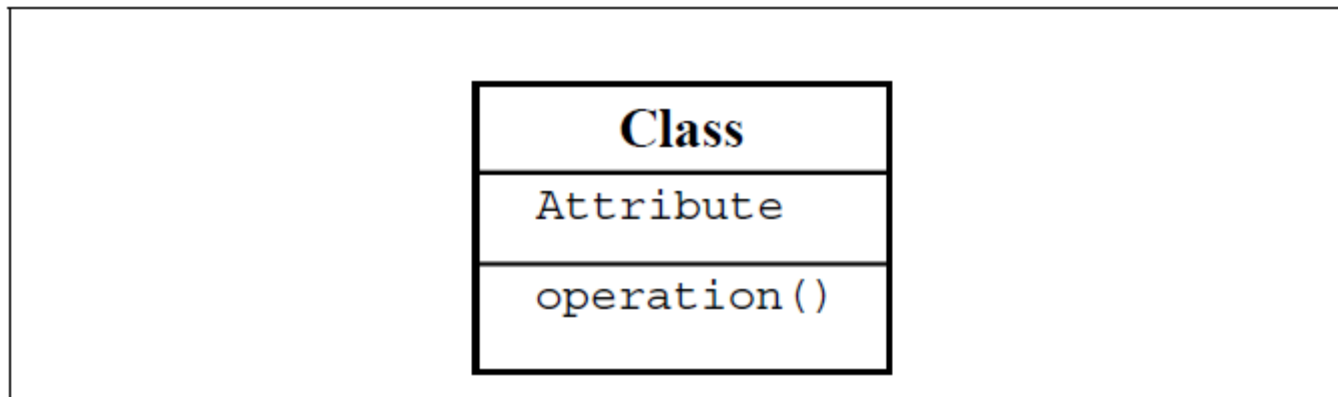
- ▶ *Class Diagram*
- ▶ *Relation*
 - ▶ *Association / Asosiasi*
 - ▶ *Generalization / Generalisasi*
 - ▶ *Aggregation / Agregasi*
 - ▶ *Composition / Komposisi*



Class Diagram

- ▶ Class diagram membantu pengembang **mendapatkan struktur sistem** dan menghasilkan **rancangan (design)** sistem yang baik
- ▶ Class diagram juga **menunjukkan property** yang diwakili oleh *attribute* dan *behaviour* melalui *method / operasi* sebuah Class

Figure 1: The Class Icon



Relation

- ▶ Secara umum hubungan/relasi antar class, meliputi:
 - ▶ *Association / Asosiasi*
 - ▶ *Generalization / Generalisasi*
 - ▶ *Aggregation / Agregasi*
 - ▶ *Composition / Komposisi*



Association / Asosiasi

- ▶ Sebuah asosiasi secara formal didefinisikan sebagai relasi antara dua atau lebih kelas menggambarkan sekelompok hubungan struktur
- ▶ Sebuah asosiasi menyiratkan bahwa obyek dari satu kelas yang memanfaatkan sebuah objek dari kelas lain dan ditunjukkan hanya dengan garis yang solid menghubungkan ikon dua kelas

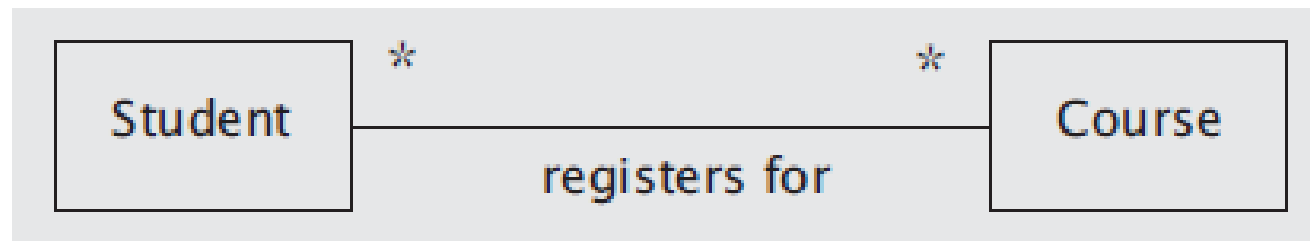


Figure 3.1 *Association between classes*

Association / Asosiasi

Example of Association

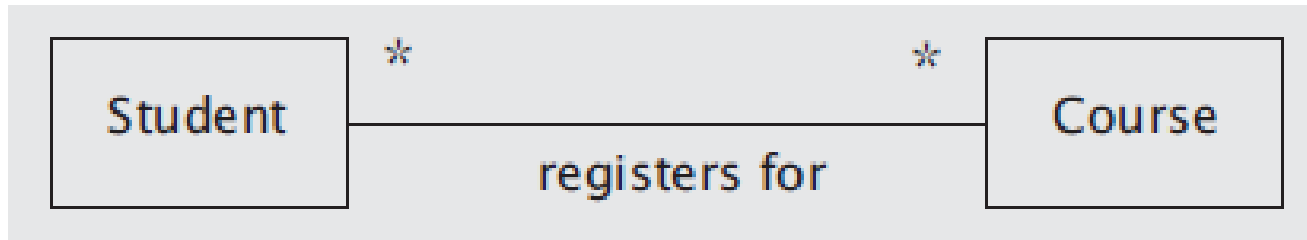


Figure 3.1 *Association between classes*

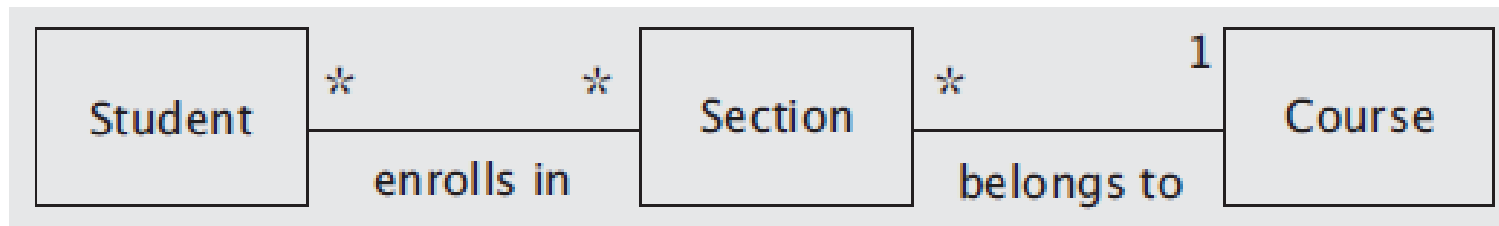


Figure 3.2 *Association involving three classes*



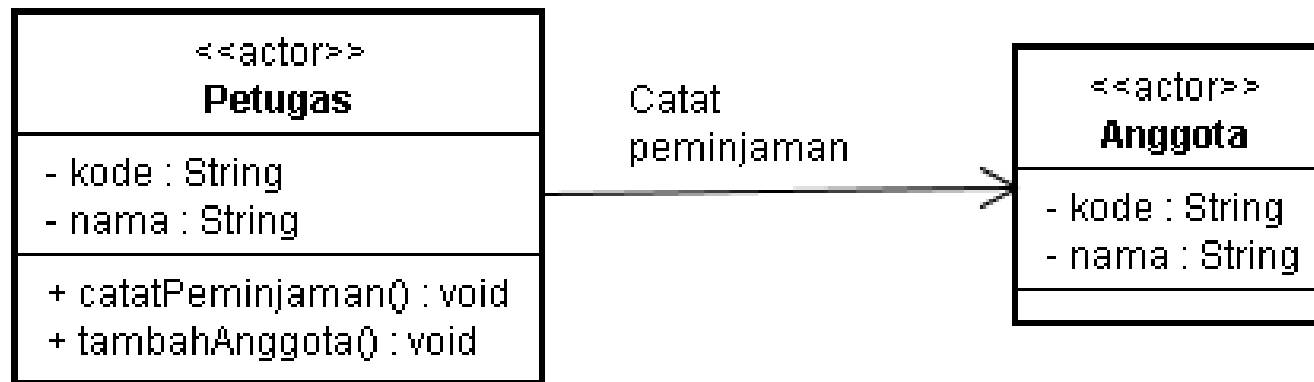
Association / Asosiasi

- ▶ **Asosiasi ada beberapa jenis, antara lain:**
 - ▶ *Directional Association / Asosiasi 1 arah*
 - ▶ *Bidirectional Association / Asosiasi 2 arah*



Directional Association / Asosiasi 1 arah

- ▶ Asosiasi ini menggambarkan bahwa pesan atau urutan kejadian terjadi hanya dari salah satu kelas sedangkan kelas yang lain pasif (dari kelas sumber ke kelas tujuan)

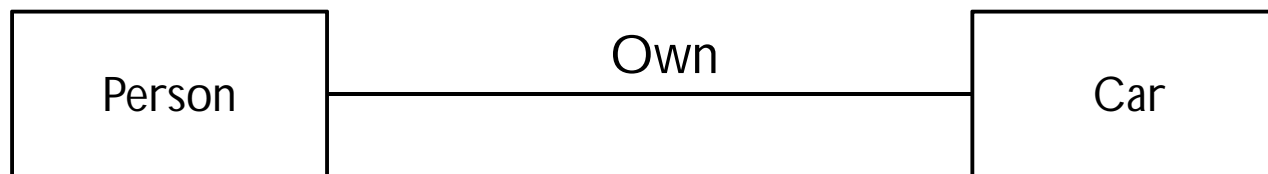


Asosiasi 1 arah



Bidirectional Association / Asosiasi 2 arah

- ▶ Asosiasi ini terjadi ketika salah satu kelas mengirimkan pesan kepada kelas yang lain kemudian kelas yang lain mengirimkan pesan kepada kelas yang mengirimnya pesan



Asosiasi Bidireksional



Generalization / Generalisasi

- ▶ *Inheritance* dapat didefinisikan sebagai mekanisme yang disediakan dalam bahasa pemrograman untuk mencapai gagasan generalisasi vertikal

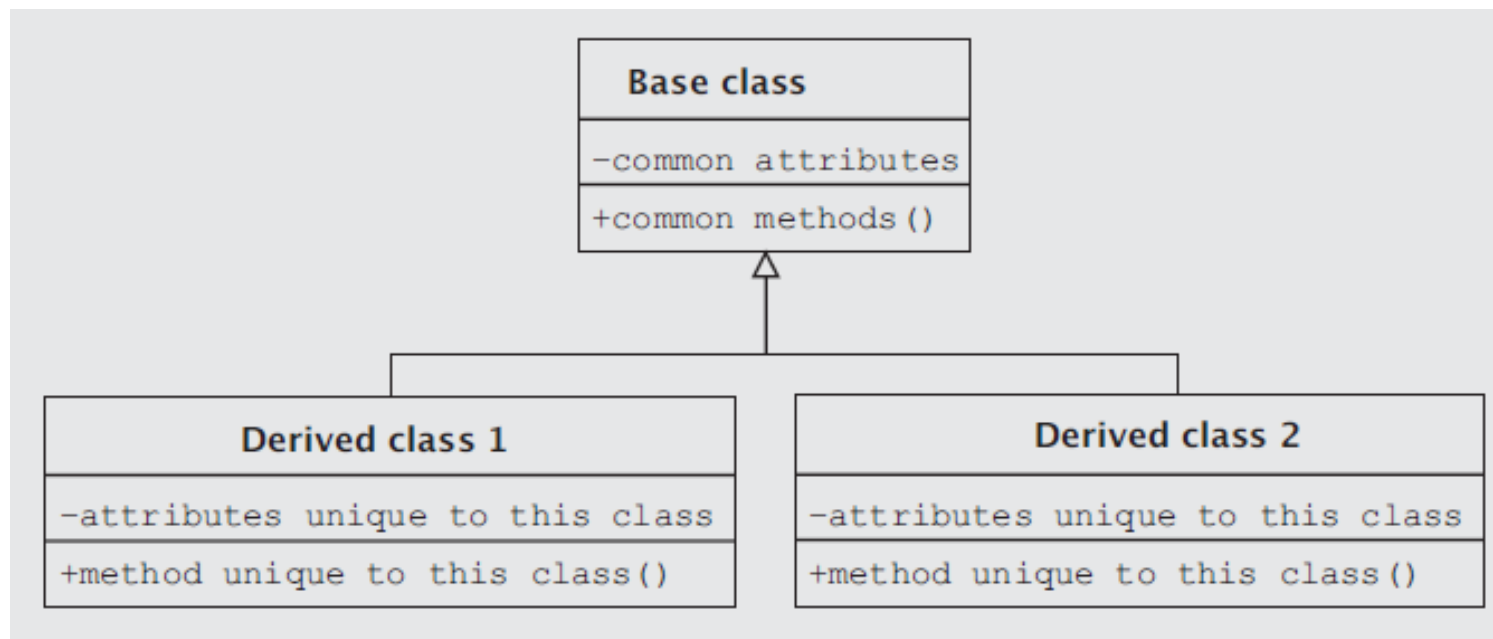


Figure 3.5 *Basic idea of inheritance*

Generalization / Generalisasi

- ▶ *Example of Generalization*

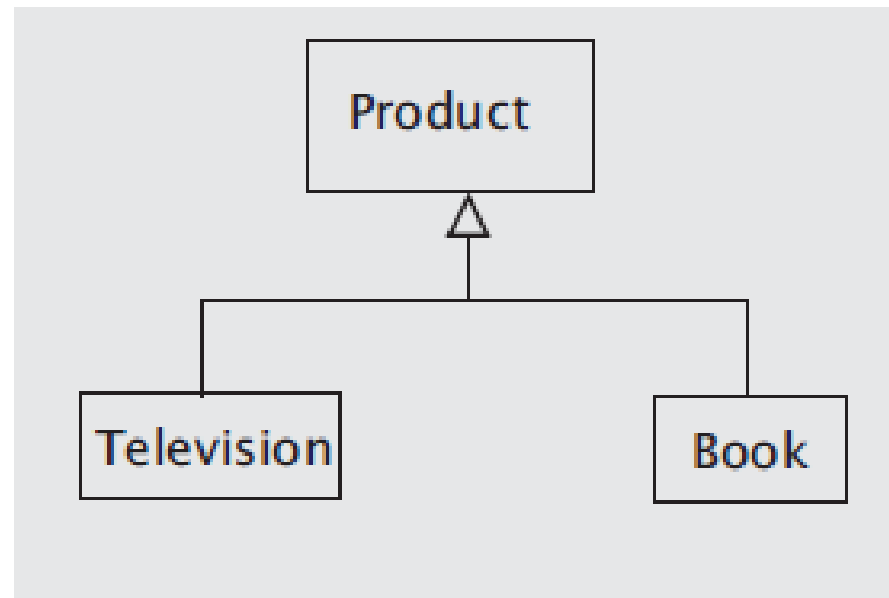


Figure 3.6 *Inheriting from product*



Generalization / Generalisasi

▶ *Example of Generalization*

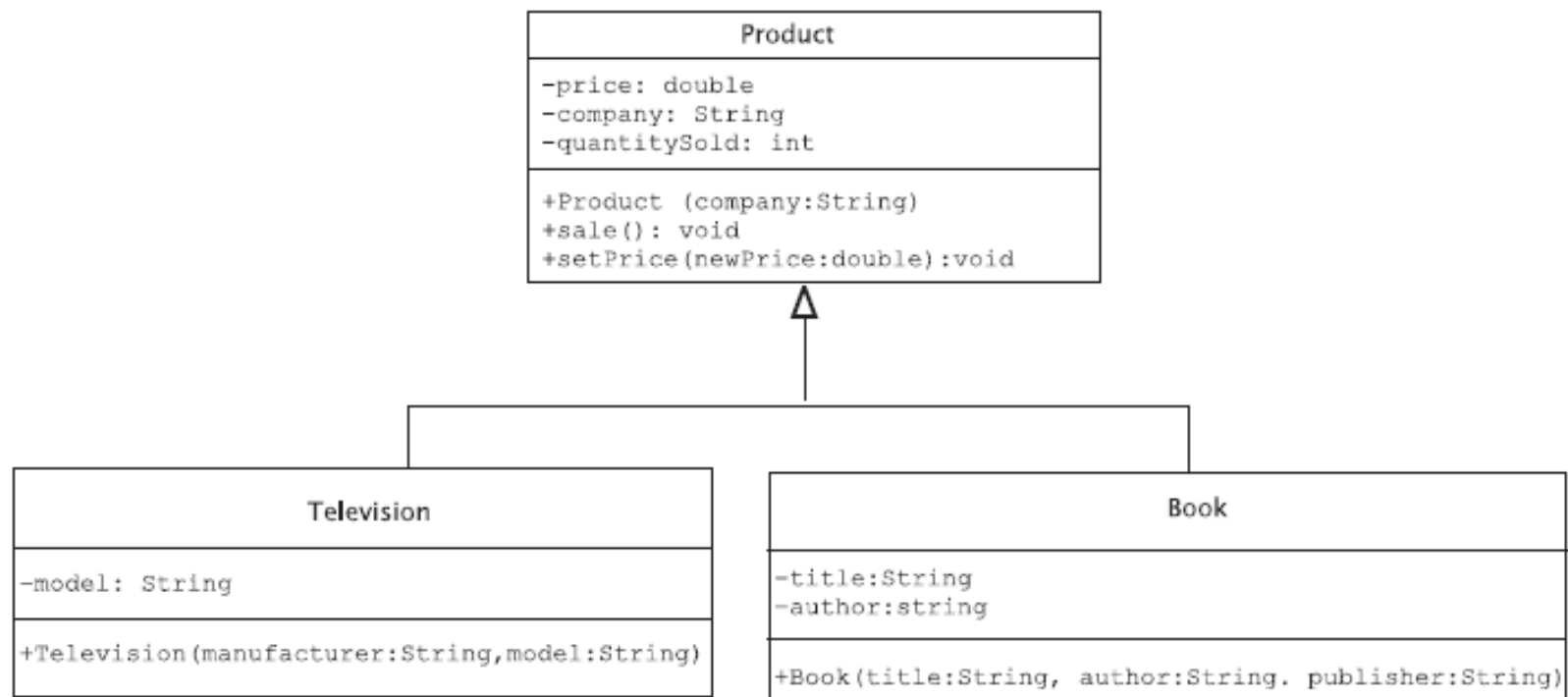


Figure 3.8 *Inheriting from product*



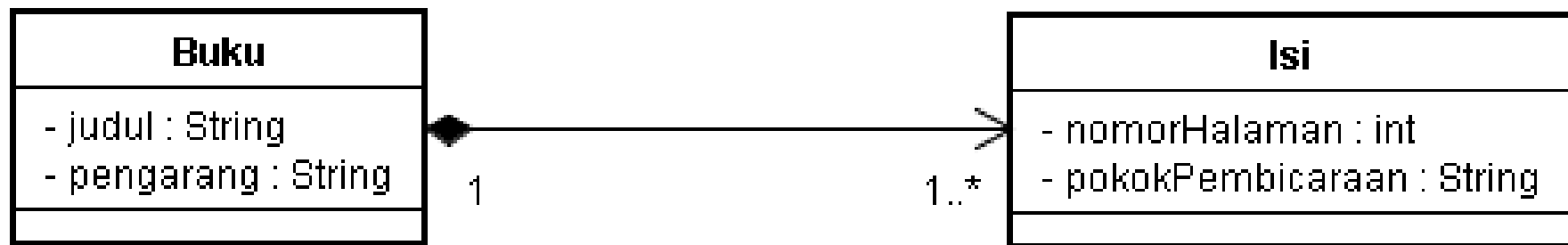
Aggregation / Agregasi

- ▶ Agregasi dapat diartikan bahwa suatu kelas merupakan bagian dari kelas yang lain namun bersifat tidak wajib
- ▶ Contohnya sebuah buku memiliki pengarang, judul, namun bisa saja suatu buku tidak memiliki daftar pustaka. Namun tetap sebagai buku.



Composition / Komposisi

- ▶ Komposisi diartikan bahwa suatu kelas merupakan bagian yang wajib dari kelas yang lain. Eksistensi (keberadaan) kelas tersebut, mempengaruhi keberadaan kelas yang lain
- ▶ Contoh kasus yaitu pada sebuah buku, sudah pasti terdapat halaman isi sekurang-kurangnya satu



Multiplicity

- ▶ Pada relasi terdapat suatu penanda yang disebut **multiplicity**.
- ▶ Multiplicity ini akan mengindikasikan berapa banyak obyek dari suatu kelas terelasi ke obyek lain
- ▶ Multiplicity memiliki batas bawah dan batas atas
 - ▶ Jika batas bawah dan batas atas bernilai sama yaitu 1..1, maka hanya ditulis 1
 - ▶ Sedangkan * merupakan kependekan dari 0..*
 - ▶ Dan 0 merupakan kependekan dari 0..0
- ▶ Default nilai multiplicity relasi class diagram adalah 1



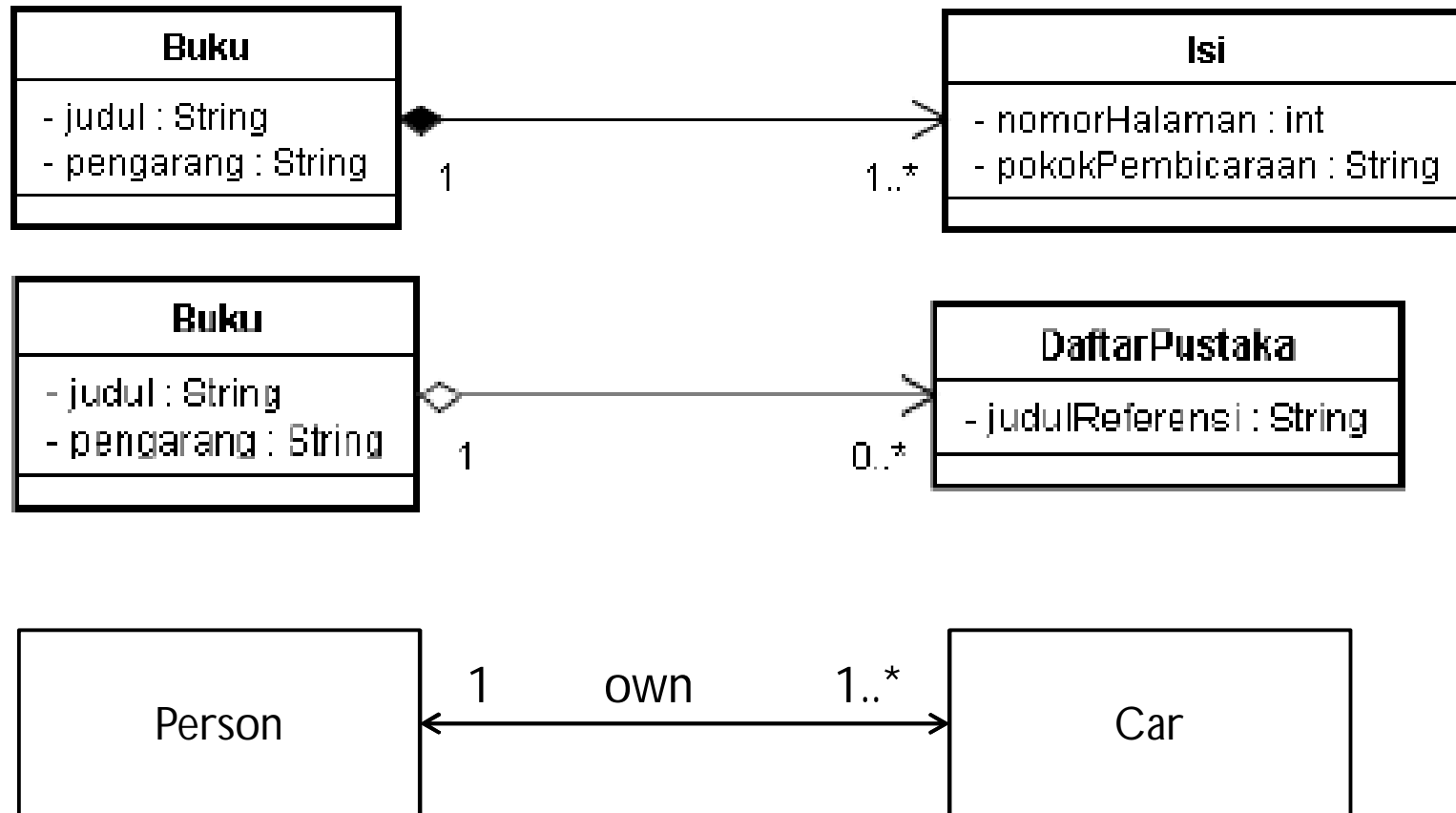
Multiplicity

Multiplicity	Arti
* = 0..*	Banyak/tak terhingga
0 = 0..0	Nol
1 = 1..1	Tepat Satu, bisa ditulis bisa tidak (<i>default</i>)
1..*	Antara Satu sampai Banyak
0..1	Nol atau Satu

Notasi UML Class Diagram untuk Multiplicity



Multiplicity - Contoh



Thanks

&

See You

Next Chapter

