



Kuliah
Rekayasa Perangkat Lunak
(Software Engineering)
Bagian 2

Software Engineering: A Practitioner's Approach, 6/e

Chapter 8
Pemodelan Analisis



Pemodelan CRC

- Class-class analisis memiliki “tanggung-jawab”
 - *Tanggungjawab* adalah atribut-atribut dan operasi-operasi yang terenkapsulasi oleh class
- Class-class analisis berkolaborasi satu dengan yang lain
 - *Collaborators* adalah class-class yang dibutuhkan untuk menyediakan sebuah class dengan informasi yang dibutuhkan untuk memenuhi tanggung jawabnya.
 - Secara umum, sebuah kolaborasi berakibat permintaan informasi atau permintaan beberapa aksi/operasi.

Pemodelan CRC



Class: FloorPlan	
Description:	
Responsibility:	Collaborator:
defines floor plan name/type	
manages floor plan positioning	
scales floor plan for display	
scales floor plan for display	
incorporates walls, doors and windows	Wall
shows position of video cameras	Camera

Tipe-tipe Class

- **Class entitas**, sering disebut class model atau bisnis, yang diekstrak langsung dari statemen permasalahan (contoh : Sensor).
- **Class perbatasan** digunakan untuk membuat interface (contoh : layar interaktif, atau laporan cetak) dimana user melihat dan berinteraksi dengannya selama PL digunakan.
- **Class kendali** mengelola “unit kerja [UML03] dari awal sampai akhir. Class kendali dapat didesain mengelola :
 - Pembuatan atau update objek entitas;
 - Inisiasi objek perbatasan sebagaimana mereka mendapatkan informasi dari objek entitas;
 - Komunikasi kompleks antara sekumpulan objek;
 - Validasi data yang dikomunikasikan antara user dan aplikasi.



Responsibilities

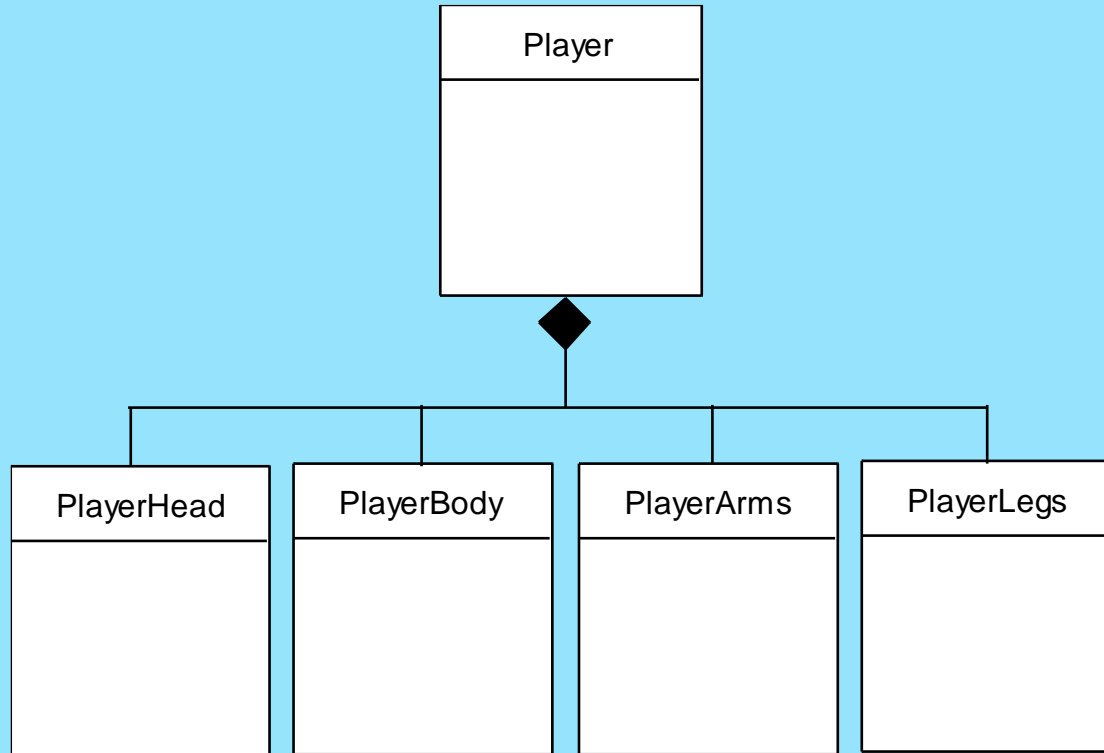
- System intelligence should be distributed across classes to best address the needs of the problem
- Each responsibility should be stated as generally as possible
- Information and the behavior related to it should reside within the same class
- Information about one thing should be localized with a single class, not distributed across multiple classes.
- Responsibilities should be shared among related classes, when appropriate.

Kolaborasi



- Class memenuhi tanggung jawabnya dengan satu diantara dua cara :
 - Sebuah class dapat menggunakan operasinya sendiri untuk memanipulasi atributnya masing-masing atau
 - Sebuah class dapat berkolaborasi dengan class lainnya.
- Kolaborasi membuat relasi antara class
- Kolaborasi dapat diidentifikasi dengan menentukan apakah sebuah class dapat memenuhi tanggung jawabnya masing-masing
- Tiga relasi umum yang berbeda antar class [WIR90]:
 - *is-part-of* relationship
 - *has-knowledge-of* relationship
 - *depends-upon* relationship

Composite Aggregate Class



Review model CRC



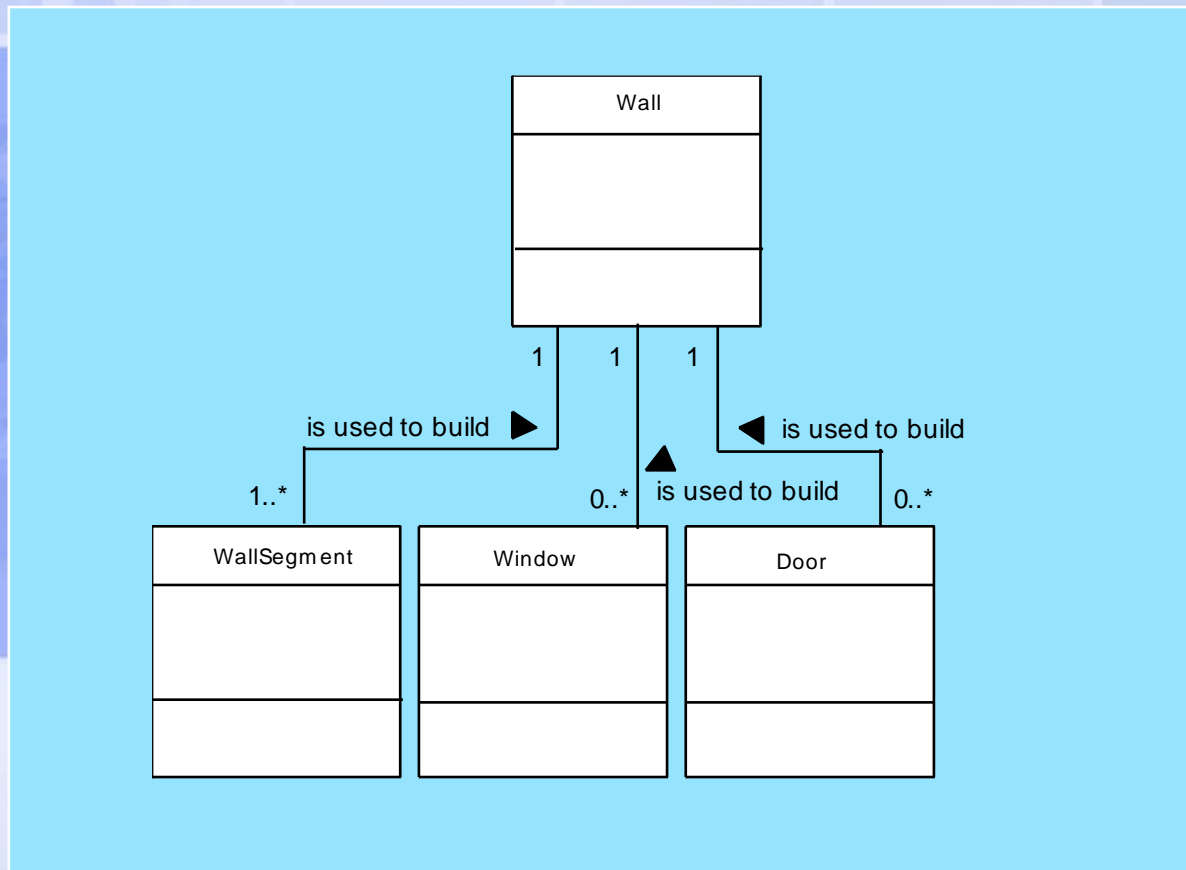
- Semua peserta dalam review (model CRC) diberikan sebuah subset dari kartu index model CRC.
 - Kartu yang berkolaborasi harus terpisah (tidak boleh ada reviewer yang memiliki dua kartu yang berkolaborasi).
- Semua skenario use-case (dan diagram use case terkait) harus diorganisasi dalam kategori-kategori.
- Pemimpin review membaca use-case secara hati-hati.
 - Ketika pemimpin review sampai pada objek, dia akan memberi tanda kepada person yang memegang kartu index class yang terkait.
- Ketika tanda dikirimkan, pemilik kartu class diminta untuk menggambarkan tanggung jawab yang tertulis di kartu tersebut.
 - Kelompok menentukan satu (atau lebih) tanggung jawab yang memenuhi kebutuhan use-case.
- Jika tanggung jawab dan kolaborasi yang tertera pada kartu index tidak dapat mengakomodasi use-case, modifikasi dilakukan pada kartu tersebut.
 - Hal ini termasuk definisi class baru (dan kartu index CRC) atau spesifikasi baru atau revisi mengenai tanggung jawab, atau kolaborasi kartu yang sudah ada.



Asosiasi dan Dependensi

- Dua class analisis sering berhubungan satu dengan yang lain dalam beberapa pola
 - Dalam UML relasi ini sering disebut *asosiasi*
 - Asosiasi dapat didapatkan dengan mengenali *multiplicity* (istilah *cardinality* digunakan dalam pemodelan data)
- Dalam banyak instans, relasi client-server ada diantara dua class analisis.
 - Dalam kasus ini, class client tergantung pada class server dalam suatu cara dan **relasi dependensi** terjadi

Multiplicity



Dependencies

