
Use Case, Class, Object Diagram

SI-215

Analisa & Desain Sistem Informasi I

Rosa Ariani Sukamto, ST

Perlunya Permodelan

- Permodelan adalah gambaran dari realita yang simpel
- Permodelan digunakan untuk mempermudah langkah berikutnya dari pengembangan SI
- Permodelan digunakan untuk mengurangi resiko
- Permodelan digunakan untuk memvisualkan SI yang akan dibuat

Unified Modeling Language (UML)

- Muncul karena adanya kebutuhan permodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak
- Merupakan bahasa visual untuk permodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung



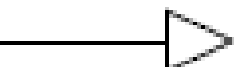
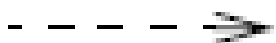
Rational Unified Process (RUP)

- SDLC berbasis use case
- Fase:
 - Inception
 - Identifikasi kebutuhan, ruang lingkup sistem, dan analisa resiko
 - Elaboration
 - Desain, implementasi sisi developer, testing
 - Construction
 - Implementasi sistem lengkap, instalasi
 - Transition
 - Memastikan software memenuhi kebutuhan user (maintenance)

Use Case

- Merupakan permodelan untuk kelakuan (*behavior*) sistem
- Mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem
- Syarat penamaan pada use case: simpel
 - Aktor
 - Orang, proses, atau sistem lain yang berinteraksi dengan sistem, sub sistem, atau kelas
 - Use case
 - Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor

Relasi Use Case

- Association 
 - Komunikasi antara aktor dan use case yang berpartisipasi pada use case
`«extend»`
- Extend 
 - Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu
- Generalization 
 - Hubungan generalisasi dan spesialisasi antara dua buah use case
`«include»`
- Include 
 - Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya

Contoh Use Case

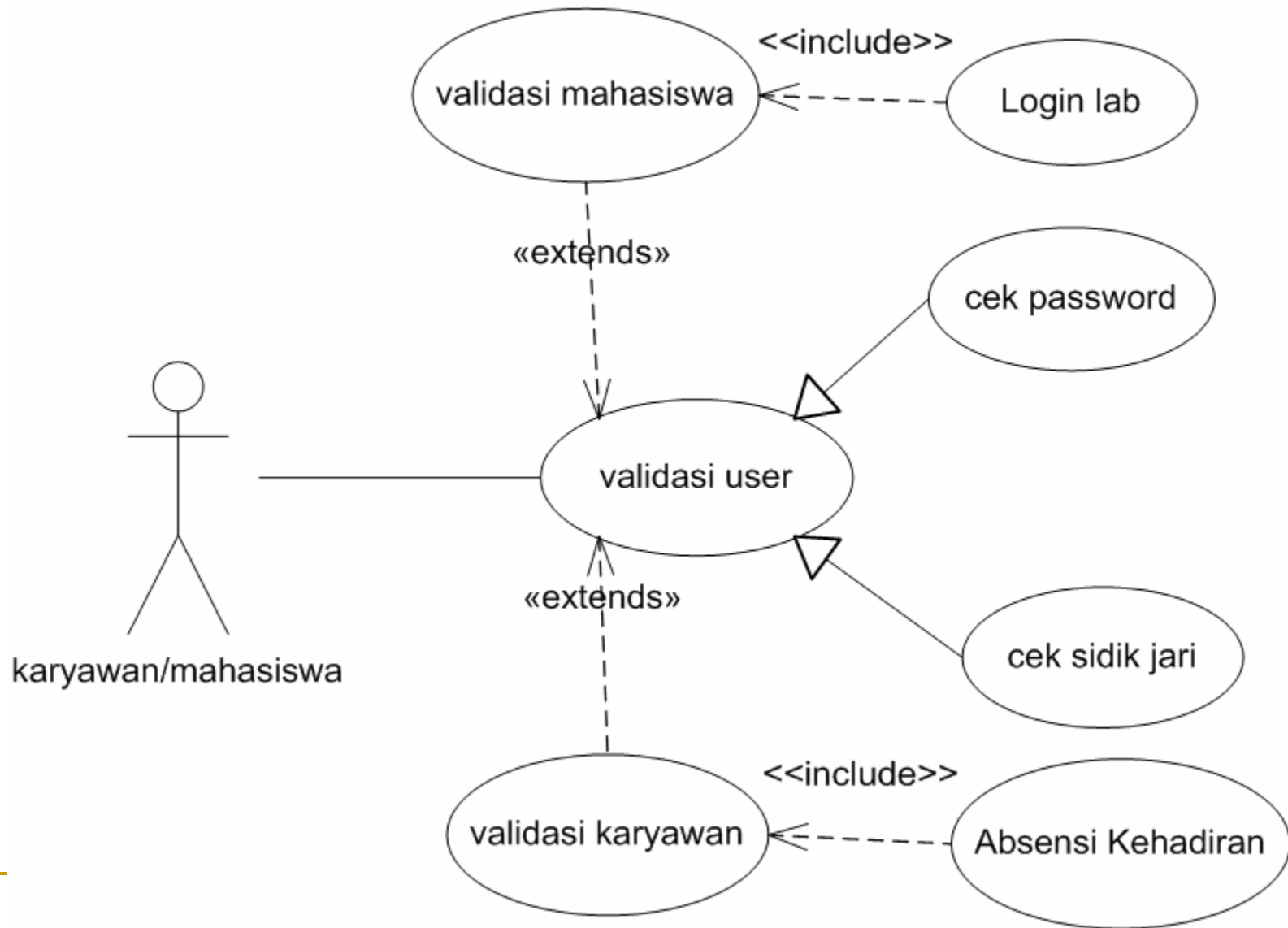
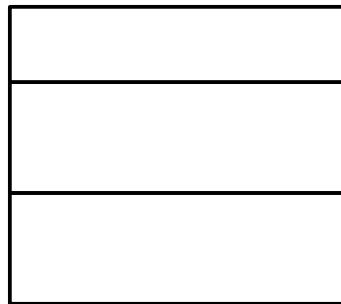


Diagram Kelas & Diagram Objek

- Menggambarkan struktur sistem
 - Diagram kelas
 - menggambarkan struktur kelas (program)
 - Diagram objek
 - Menggambarkan struktur objek (kerja program)
- Kelas
 - Bentuk pada kode program (source code)
- Objek
 - Kelas pada kode program dieksekusi menjadi objek (program yang dijalankan)

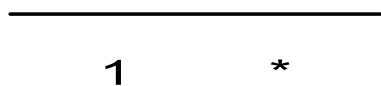
Diagram Kelas & Diagram Objek

- Kelas

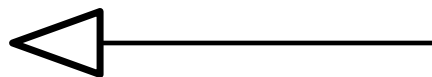


- Relasi

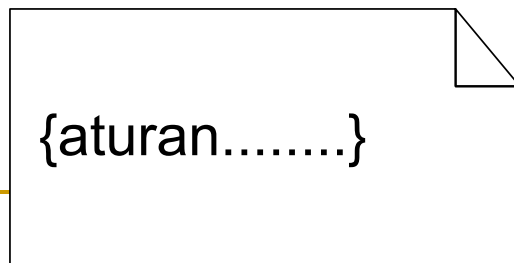
- Association



- Generalization



- Constraint



Nama_kelas

-atribut_1 : int

-atribut_2 : float

+getAtribut() : int

+setAtribut()

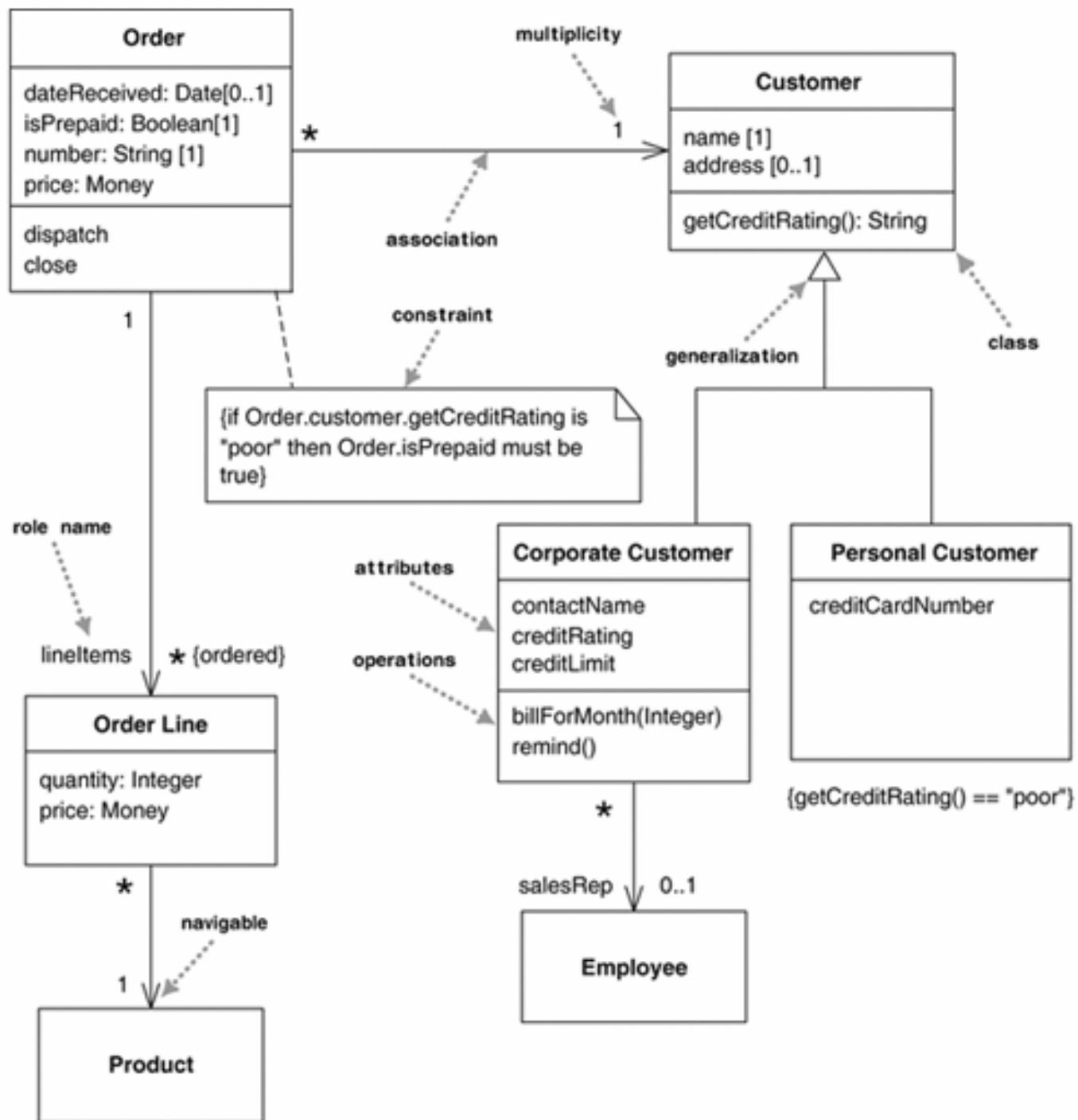


Diagram Kelas & Diagram Objek

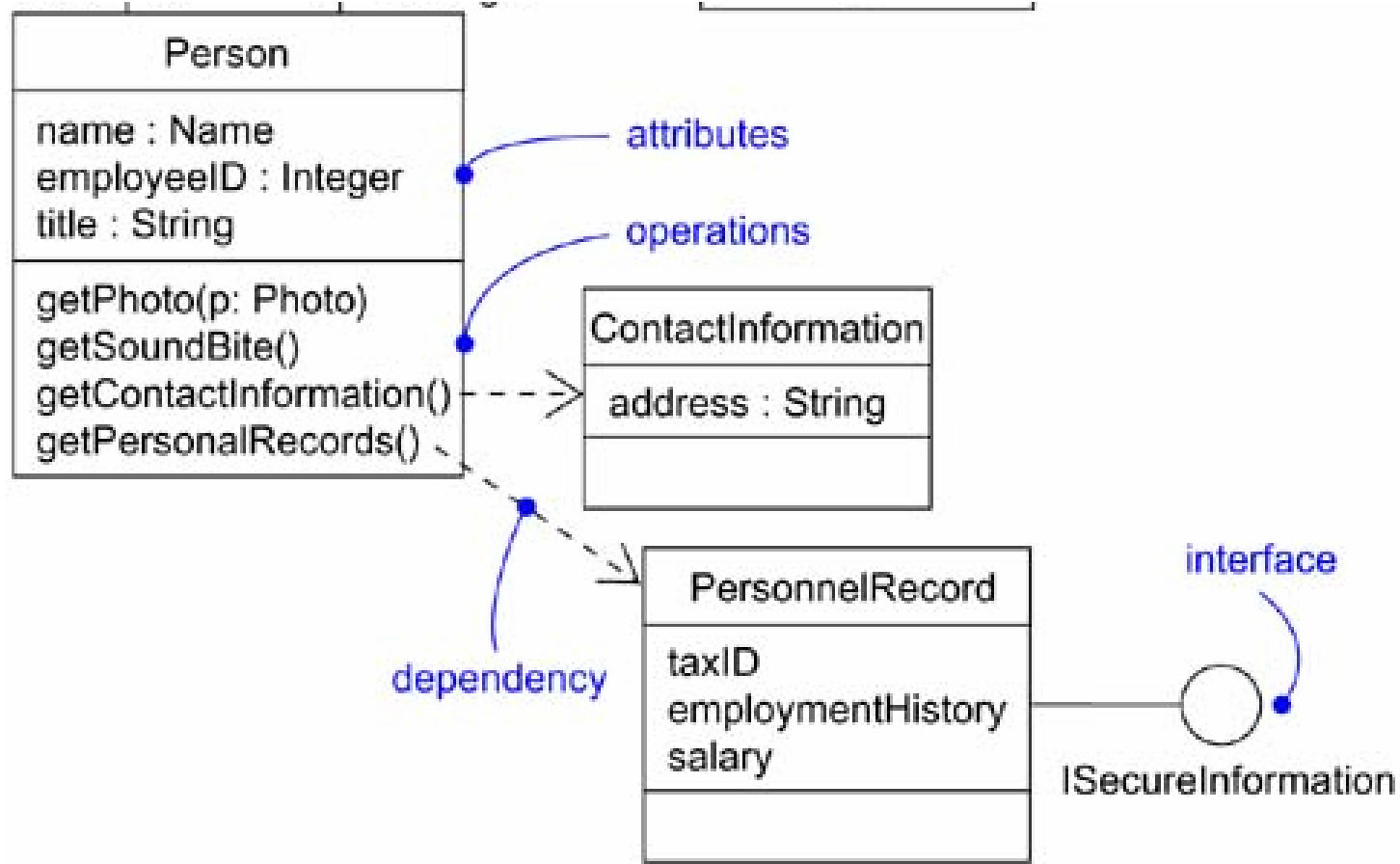
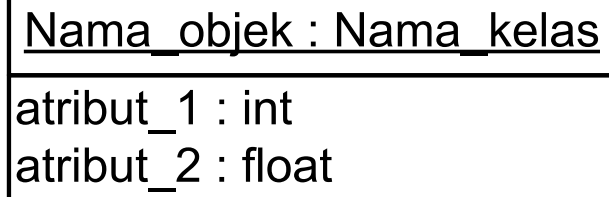


Diagram Kelas & Diagram Objek

- Objek



- Link

Diagram Kelas & Diagram Objek

