PEMODELAN PERANGKAT LUNAK BAG – 3

Telkom University

Budhi Irawan, S.Si, M.T

CLASS DIAGRAM (1)

- ***Class*** adalah sebuah spesifikasi yang jika ***di-instansiasi*** akan menghasilkan sebuah ***objek*** dan merupakan inti dari pengembangan dan ***desain berorientasi objek***.
- ***Class*** menggambarkan ***keadaan (atribut/properti)*** suatu sistem, sekaligus menawarkan layanan untuk ***memanipulasi keadaan tersebut (metoda/fungsi)***.
- ***Class diagram*** menggambarkan ***struktur*** dan ***deskripsi class, package*** dan ***objek*** beserta hubungan satu sama lain seperti ***pewarisan, asosiasi,*** dan ***lain-lain***.

CLASS DIAGRAM (2)

- ***Diagram Class*** memberikan pandangan secara luas dari suatu sistem dengan menunjukan ***kelas-kelasnya*** dan hubungan mereka. Diagram Class bersifat ***statis***; *menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan*.
- ***Class diagram*** dapat membantu dalam ***memvisualisasikan struktur kelas-kelas*** dari suatu sistem dan merupakan tipe diagram yang paling ditemui dalam pemodelan system berbasis ***object-oriented***.

OBJECT

- **Object** adalah gambaran dari ***entity***, baik ***dunia nyata*** atau ***konsep*** dengan batasan-batasan yang tepat.
- ***Object*** bisa mewakili sesuatu yang nyata dalam ***domain problem*** kita seperti ***komputer, barang, konsumen***, dapat berupa konsep seperti ***proses penarikan uang, pembayaran, pengembalian buku*** dan lain-lain.
- Dari ***object-object*** ini kita bisa mengabstraksikan ***candidate class*** yang mungkin terlibat.

KARAKTERISTIK OBJECT

1. **State,** merupakan suatu ***kondisi / keadaan*** dari ***object*** yang mungkin ada. **Status dari object** akan berubah setiap waktu dan ditentukan oleh sejumlah **property** dan **relasi** dengan **object** lainnya.
2. **Behavior (sifat)** menentukan bagaimana ***object*** merespon permintaan dari ***object*** lain dan melambangkan setiap hal yang dapat dilakukan.
3. **Identity (identitas)** artinya setiap ***object*** yang ada dalam suatu system adalah **“unik”**.

CARA MENEMUKAN OBJECT (1)

1. Pengelompokan berdasarkan ***kata/frasa*** benda pada ***skenario*** / ***dokumentasi use case***
2. Berdasarkan daftar kategori ***objek***, antara lain:
   - Objek fisik: pesawatTelepon
   - Spesifikasi/rancangan/deskripsi: deskripsiPesawat
   - Tempat: gudang
   - Transaksi: penjualan
   - Butir yang terlibat pada transaksi: barang jualan

CARA MENEMUKAN OBJECT (2)

- Peran: pelanggan
- Wadah: pesawatTerbang
- Benda yang diwadahi: penumpang
- Organisasi: departemen
- Kejadian: pendaratan
- Proses: reservasi
- Aturan atau kebijakan: aturanDiskon
- Katalog atau rujukan: daftarPelanggan

CANDIDATE CLASS

- ***Candidate class*** dapat kita tentukan dengan melihat ***skenario use case*** yang telah kita buat. ***Candidate class*** tersebut dapat diambil dari kata **benda** yang muncul pada ***skenario use case***.

Barang

Pembayaran

Penjualan

Kasir

Struk

CLASS

- ***Class*** adalah deskripsi **sekelompok *object*** dari **property (atribut)**, sifat **(operasi)**, **relasi antar *object*** dan **semantik** yang umum.
- ***Class*** merupakan ***blueprint/ template /*cetakan** dari satu atau lebih ***object***.

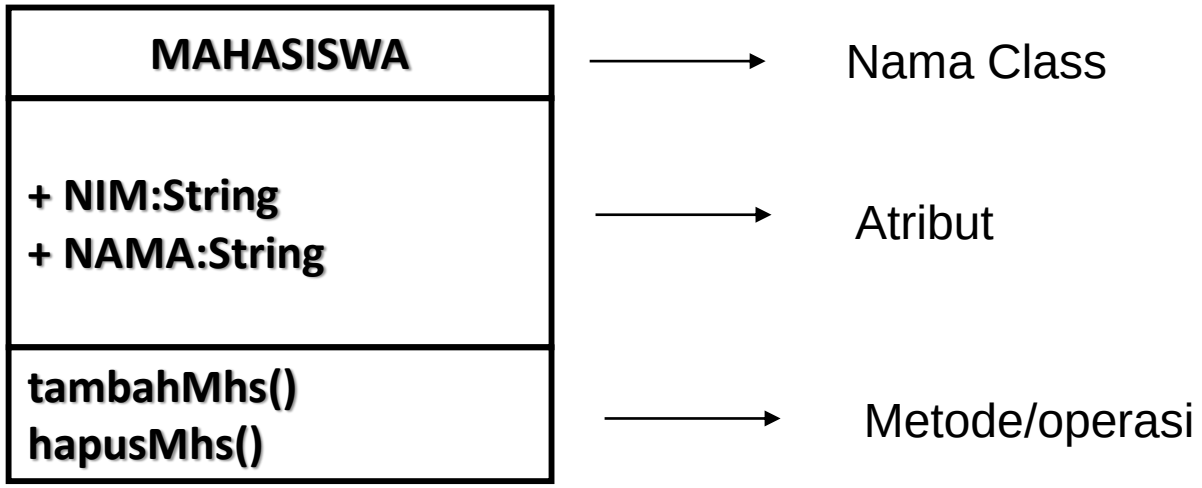
NOTASI CLASS

- Penamaan ***class*** menggunakan **kata benda tunggal** yang merupakan **abstraksi** yang terbaik.
- Pada **UML**, ***class*** digambarkan dengan **segi empat yang dibagi**.
- **Bagian atas** merupakan **nama dari *class***. **Bagian yang tengah** merupakan **struktur dari *class* (atribut)** dan **bagian bawah** merupakan **sifat dari class (operasi)**.

STRUKTUR CLASS

- *Class* memiliki **tiga area pokok** :*Nama (dan stereotype), Atribut, dan Metoda (operasi)*
- Atribut dan metoda dapat memiliki salah satu sifat berikut :
  - ***Private***, tidak dapat dipanggil dari luar ***class*** yang bersangkutan
  - ***Protected***, hanya dapat dipanggil oleh ***class*** yang bersangkutan dan anak-anak yang mewarisinya
  - ***Public***, dapat dipanggil oleh siapa saja

STRUKTUR CLASS



PENAMAAN CLASS (1)

- Setiap ***class*** harus memiliki sebuah ***nama*** yang dapat digunakan untuk membedakannya dari ***class*** lain.
- Penamaan ***class*** menggunakan ***kata benda tunggal*** yang merupakan ***abstraksi*** yang terbaik.
- Nama kelas dapat dituliskan dengan 2 cara :
  1. Hanya menuliskan ***nama*** dari ***class*** (*simple name*).
  2. Nama kelas diberi ***prefix*** nama ***package*** letak ***class*** tersebut (***path name***).
- Penulisan ***nama kelas***, huruf pertama dari setiap kata pada ***nama kelas*** ditulis dengan menggunakan ***huruf kapital***. Contohnya, Customer dan FraudAgent.

PENAMAAN CLASS (2)

Simple Names	Path Names
Mahasiswa	**java::awt::Rectangle**
Anggota	**Bussiness Rules::FraudAgent**
Kasir	

ATTRIBUTE

- Sebuah ***class*** mungkin memiliki beberapa ***attribute*** atau tidak sama sekali.
- ***Atribut*** merepresentasikan beberapa ***property*** dari sesuatu yang kita modelkan, yang dibagi dengan semua ***object*** dari semua ***class*** yang ada.
- Contohnya, setiap **tembok** memiliki **tinggi, lebar** dan **ketebalan**
- Untuk penulisan atribut ***class***, biasanya **huruf pertama** dari tiap kata merupakan **huruf kapital**, kecuali untuk huruf awal. **Contoh : birthDate, length.**

CARA MEMBUAT ATTRIBUTE

1. Dari dokumentasi use case.
   - Contoh : “Pemakai memasukkan nama pegawai, alamat, no ktp
   - Di apotik “ Penjualan memasukkan data obat meliputi kode, nama, jenis”
2. Dari memeriksa struktur basis data

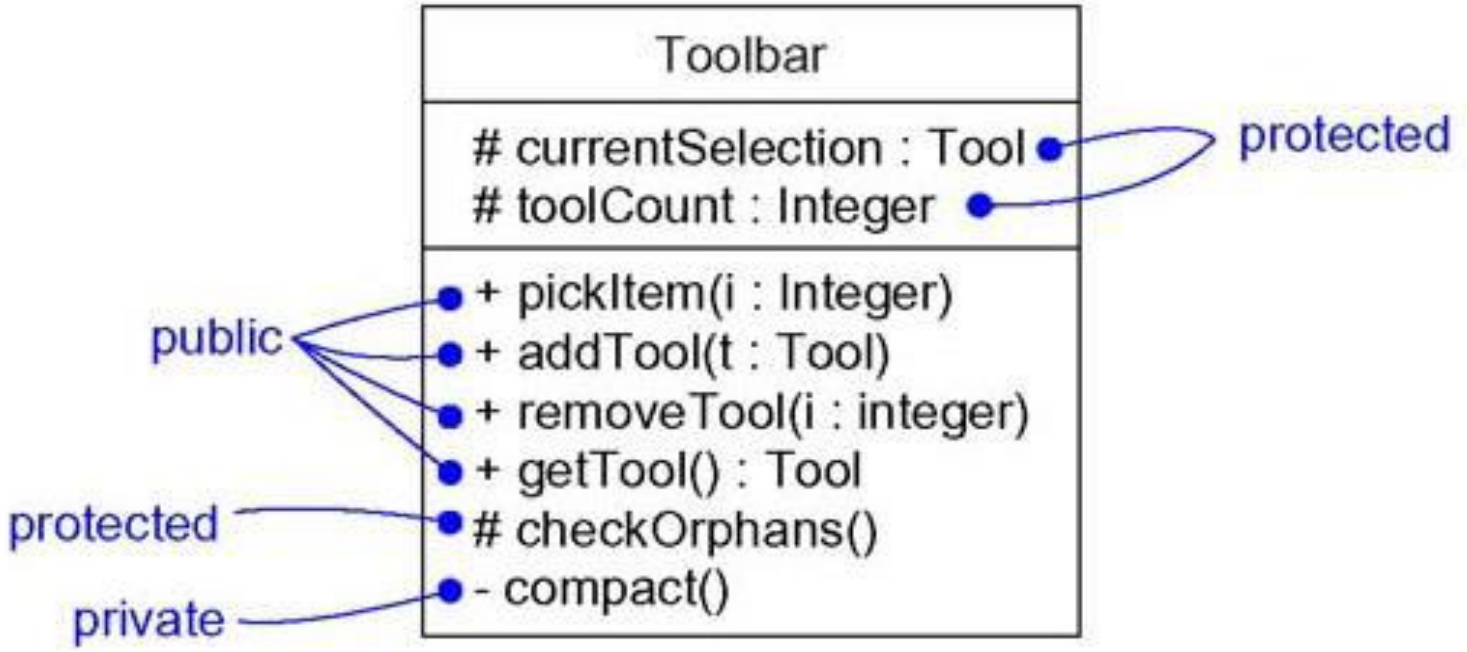
METHOD S / OPERASI

- ***Methods / Operasi*** adalah **abstraksi** dari segala sesuatu yang dapat kita lakukan pada sebuah ***object*** dan ia berlaku untuk semua ***object*** yang terdapat dalam ***class*** tersebut.
- ***Class*** mungkin memiliki beberapa ***operasi*** atau tanpa operasi sama sekali.
- Contohnya adalah sebuah ***class Kotak*** dapat ***dipindahkan, diperbesar*** atau ***diperkecil***.
- Biasanya, pemanggilan operasi pada sebuah ***object*** akan mengubah data atau kondisi dari ***object*** tersebut.

VISIBILITY SIFAT CLASS

- ***Visibility*** merupakan property yang sangat penting dalam pendefinisian atribut dan operasi pada suatu ***class***.
- ***Visibility*** menspesifikasikan apakah ***atribut/operasi*** tersebut dapat digunakan/diakses oleh ***class*** lain. UML menyediakan 3 buah tingkat *visibility*, yaitu:
  - ***Private (-)***, tidak dapat dipanggil dari luar ***class*** yang bersangkutan
  - ***Protected (#)***, hanya dapat dipanggil oleh ***class*** yang bersangkutan dan anak-anak yang mewarisinya
  - ***Public (+)***, dapat dipanggil oleh siapa saja

CONTOH SIFAT KELAS



RELATIONSHIP

- **Relasi atau** ***relationship*** menghubungkan beberapa **objek** sehingga memungkinkan terjadinya ***interaksi dan kolaborasi*** diantara ***objek-objek*** yang terhubung.
- Dalam pemodelan ***class diagram***, terdapat tiga buah ***relasi*** utama yaitu ***association, aggregation*** dan ***generalization.***

1. ASOSIASI (1)

- **Asosiasi** adalah hubungan yang terjadi antara kelas yang ada. Asosiasi memungkinkan suatu kelas untuk menggunakan atau mengetahui atribut atau operasi yang dimiliki oleh kelas lain.
- **Asosiasi** juga menggambarkan interaksi yang mungkin terjadi antara satu kelas dengan kelas yang lain. Relasi asosiasi dapat dibagi menjadi 2 (dua) jenis, yaitu

  a. uni-directional association dan

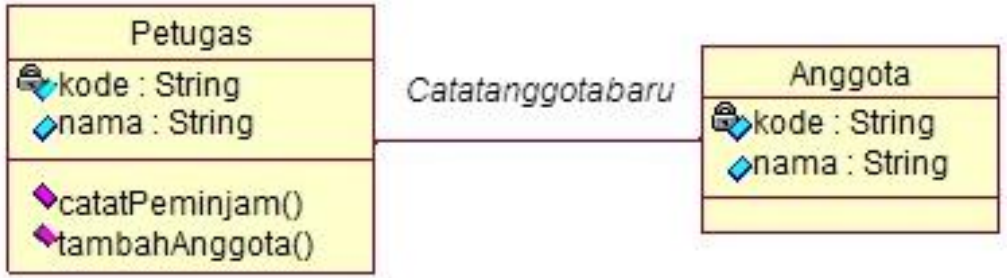
  b. bi-directional association

a. uni-directional association

- **Asosiasi** ini menggambar bahwa pesan atau urutan kejadian terjadi dari hanya salah satu kelas sedangkan kelas yang lain pasif.
- Contohnya pada saat seorang petugas perpustakaan melakukan pencatatan peminjaman terhadap seorang anggota, maka pesan dikirimkan oleh petugas dan diterima oleh anggota. Dimana petugas akan mencatat identitas anggota peminjam dan anggota peminjam berlaku pasif bukannya malah gantian mencatat identitas petugas.

a. bi-directional association

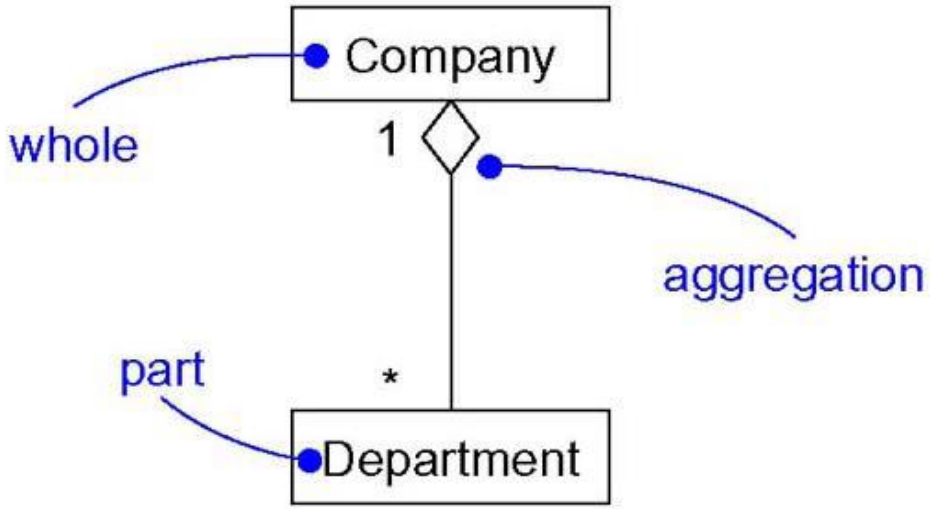
- Asosiasi ini terjadi ketika salah satu kelas mengirimkan pesan kepada kelas yang lain kemudian kelas yang lain mengirimkan pesan kepada kelas yang mengirimnya pesan.
- Contohnya pada saat seorang calon anggota m*end*aftar menjadi anggota perpustakaan maka yang terjadi adalah anggota menyerahkan identitas untuk diproses oleh petugas dan beberapa saat kemudian petugas akan memberikan kartu keanggotaan perpusatakaan.



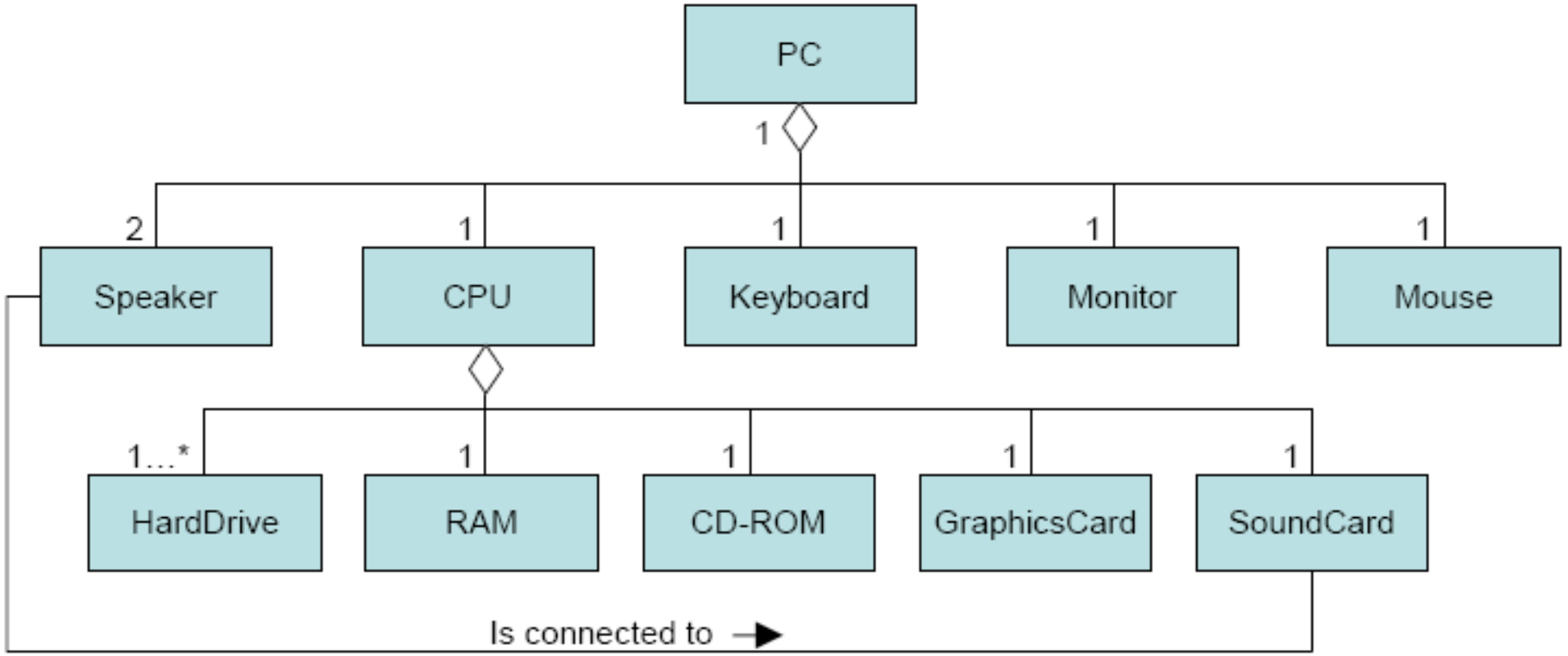
2. AGGREGATION (1)

- ***Aggregation*** merupakan bentuk khusus dari ***asosiasi*** dimana induk terhubung dengan bagian-bagiannya.
- ***Aggregation*** merepresentasikan relasi ***"has-a"***, artinya sebuah ***class*** memiliki/terdiri dari bagian-bagian yang lebih kecil.
- Dalam UML, relasi **agregasi** digambarkan dengan ***open diamond*** pada sisi yang menyatakan **induk (whole)**

2. AGGREGATION (2)



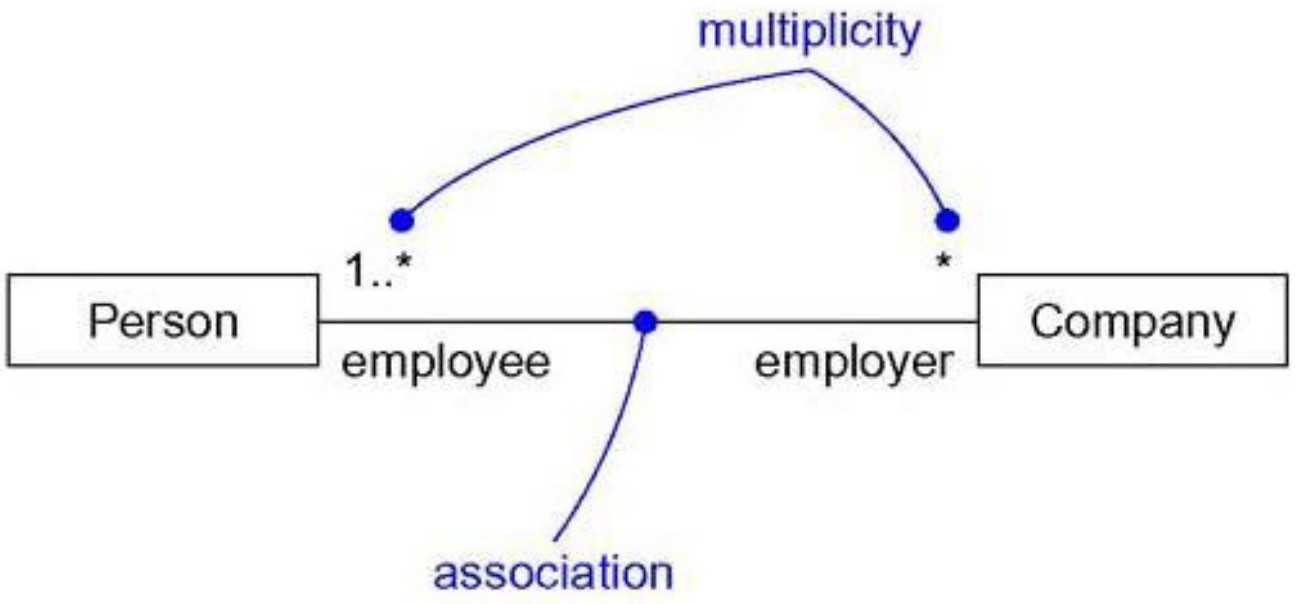
2. AGGREGATION (3)



MULTIPLICITY (1)

- ***Multiplicity*** menentukan/mendefinisikan banyaknya ***object*** yang terhubung dalam suatu relasi.
- Indikator ***multiplicity*** terdapat pada masing-masing akhir **garis relasi**, baik pada **asosiasi** maupun **agregasi**

MULTIPLICITY (2)

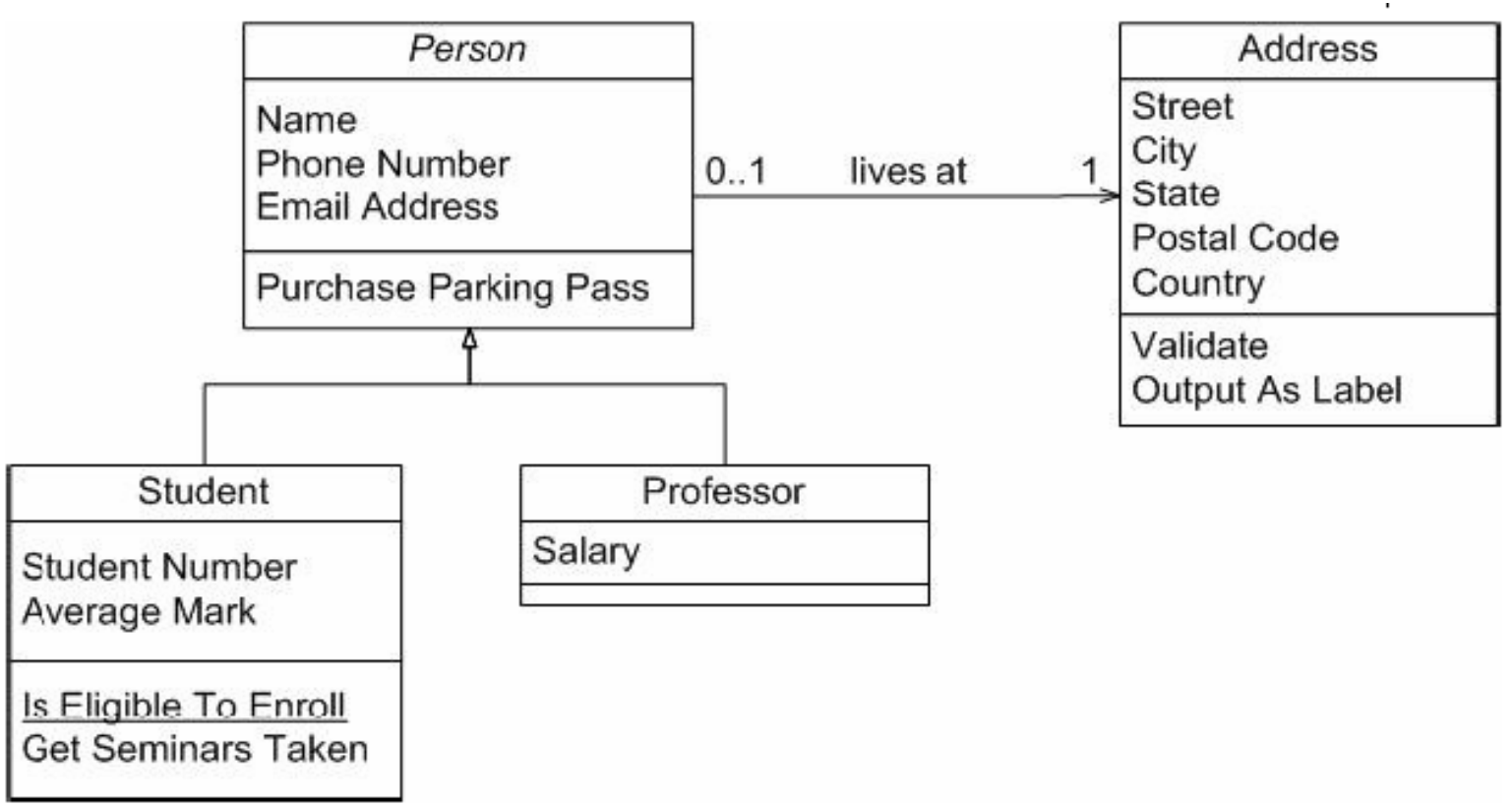


MULTIPLICITY (3)

- Pada relasi terdapat suatu penanda yang disebut ***multiplicity***.
- ***Multiplicity*** ini akan mengindikasikan berapa banyak obyek dari suatu kelas terelasi ke obyek lain.

Multiplicity	Arti
*	Banyak
0	Nol
1	Satu bisa ditulis bisa Tidak
0…*	Antara Nol sampai Banyak
1…*	Antara 1 sampai Banyak
0…1	Nol atau Satu
1…1	Tepat Satu

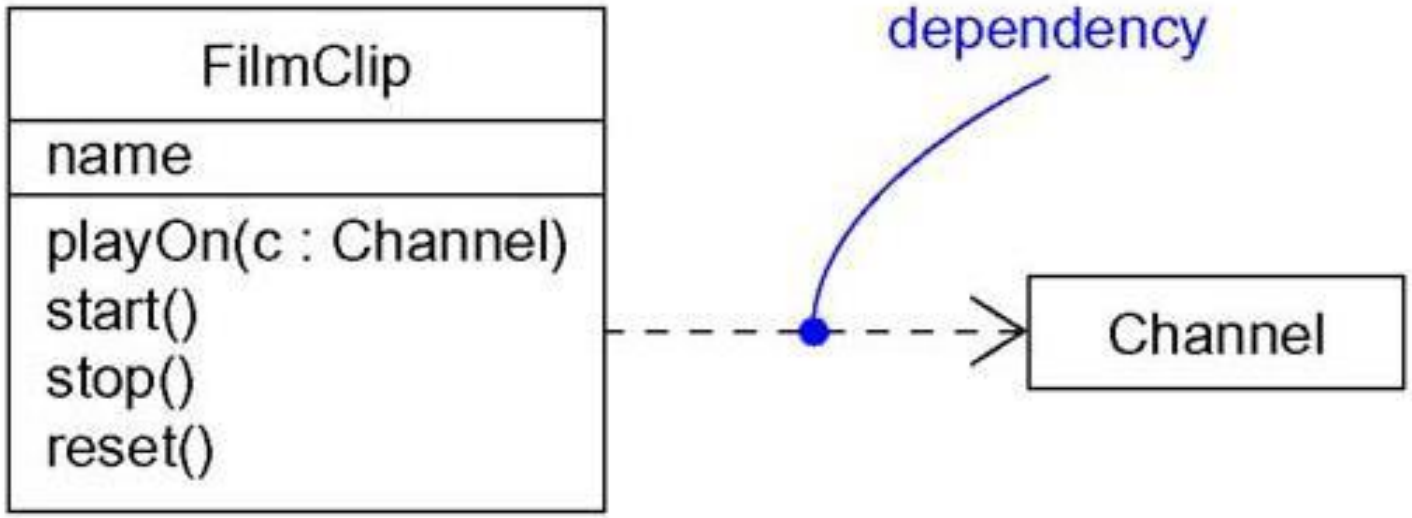
MULTIPLICITY (5)



DEPENDENCY (1)

- ***Dependency*** merupakan sebuah relasi yang menyebutkan bahwa **perubahan pada satu *class*** (misal *class* event), **maka akan mempengaruhi *class* lain yang menggunakannya** (misal *class*window), tetapi tidak berlaku sebaliknya.
- Pada umumnya, relasi ***dependency*** dalam konteks *Class* Diagram, digunakan apabila terdapat satu *class* yang menggunakan / meng-*instanceclass* lain **sebagai argumen dari sebuah method**.
- Perhatikan contoh dibawah, bila spesifikasi dari *class* Channel berubah, maka method playOn pada *class*FilmClip juga akan berubah.

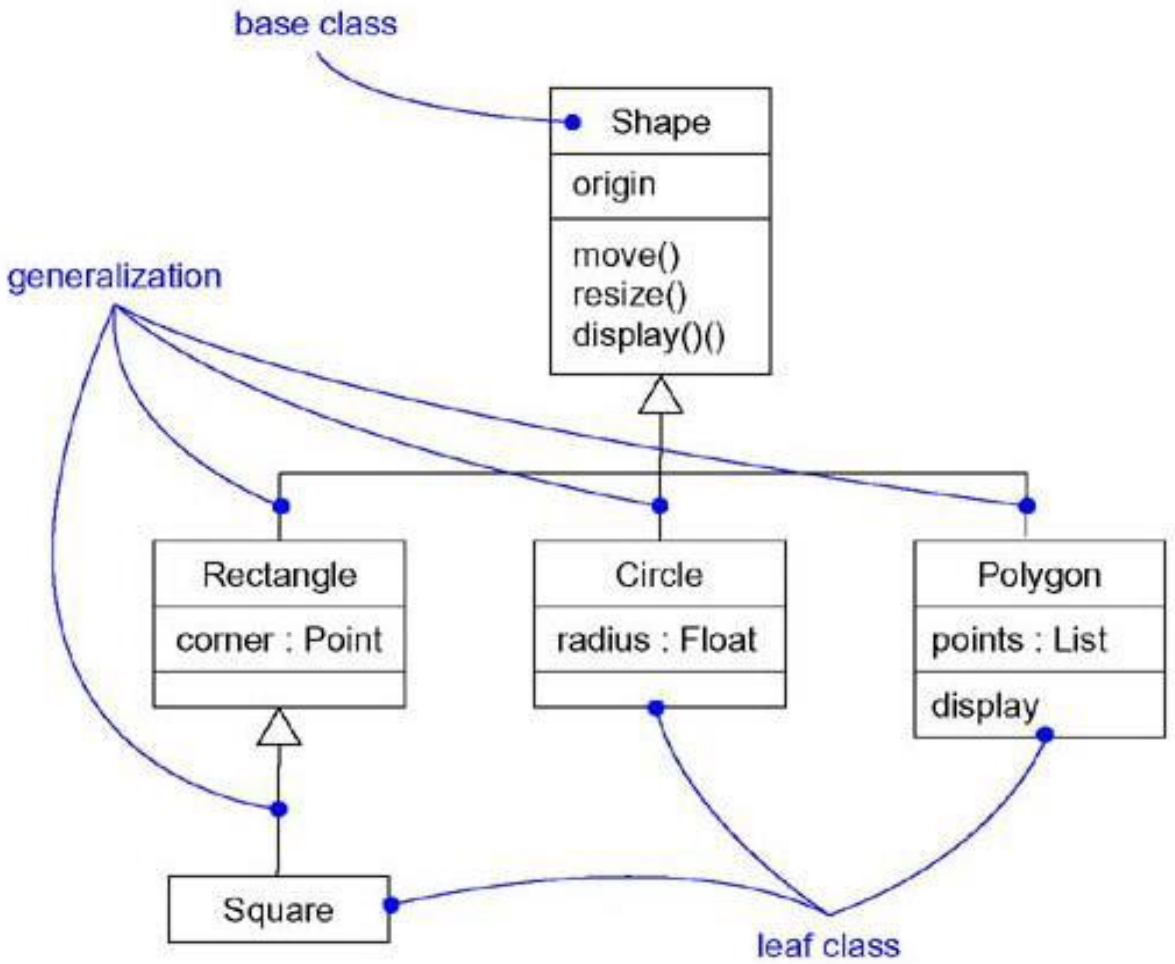
DEPENDENCY (2)



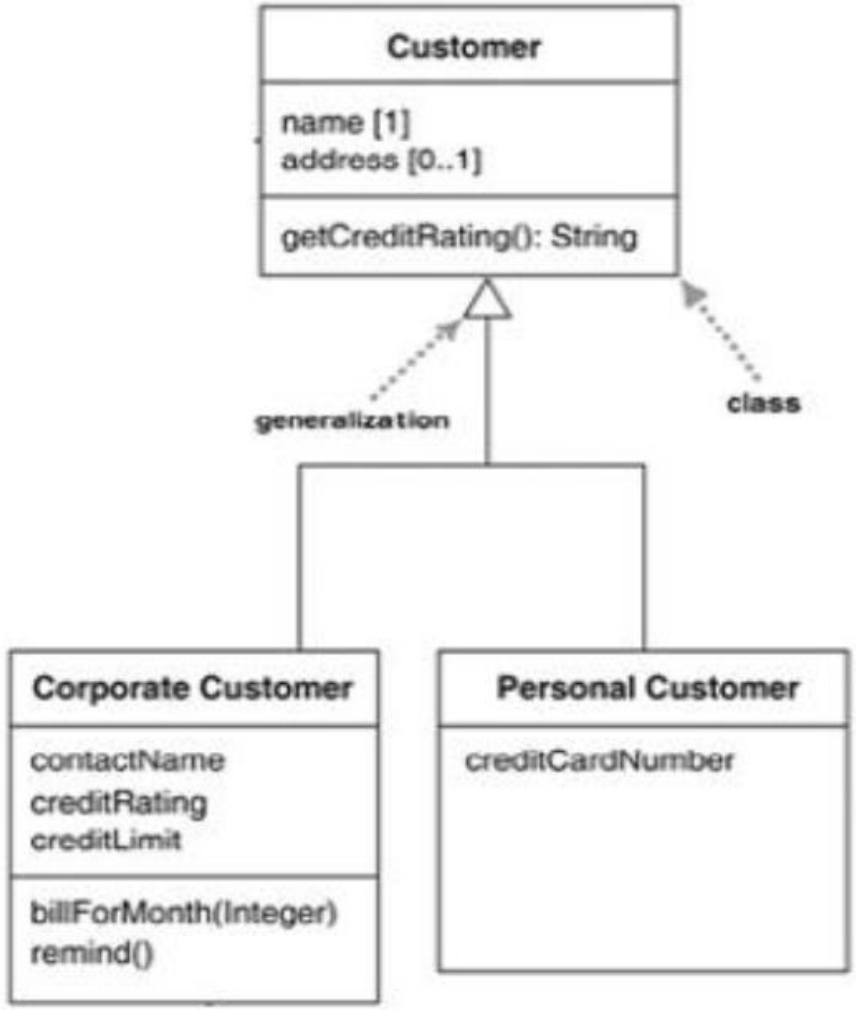
GENERALIZATION (1)

- *Generalization* menjamin kemampuan untuk membuat ***superclass*** yang melingkupi ***struktur*** dan ***behaviour*** yang umum pada beberapa ***class dibawahnya (subclass)***.

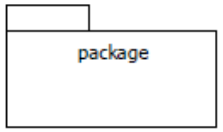
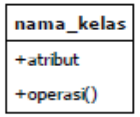
GENERALIZATION (2)



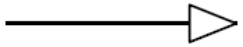
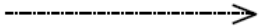
GENERALIZATION (3)



SIMBOL CLASS DIAGRAM (1)

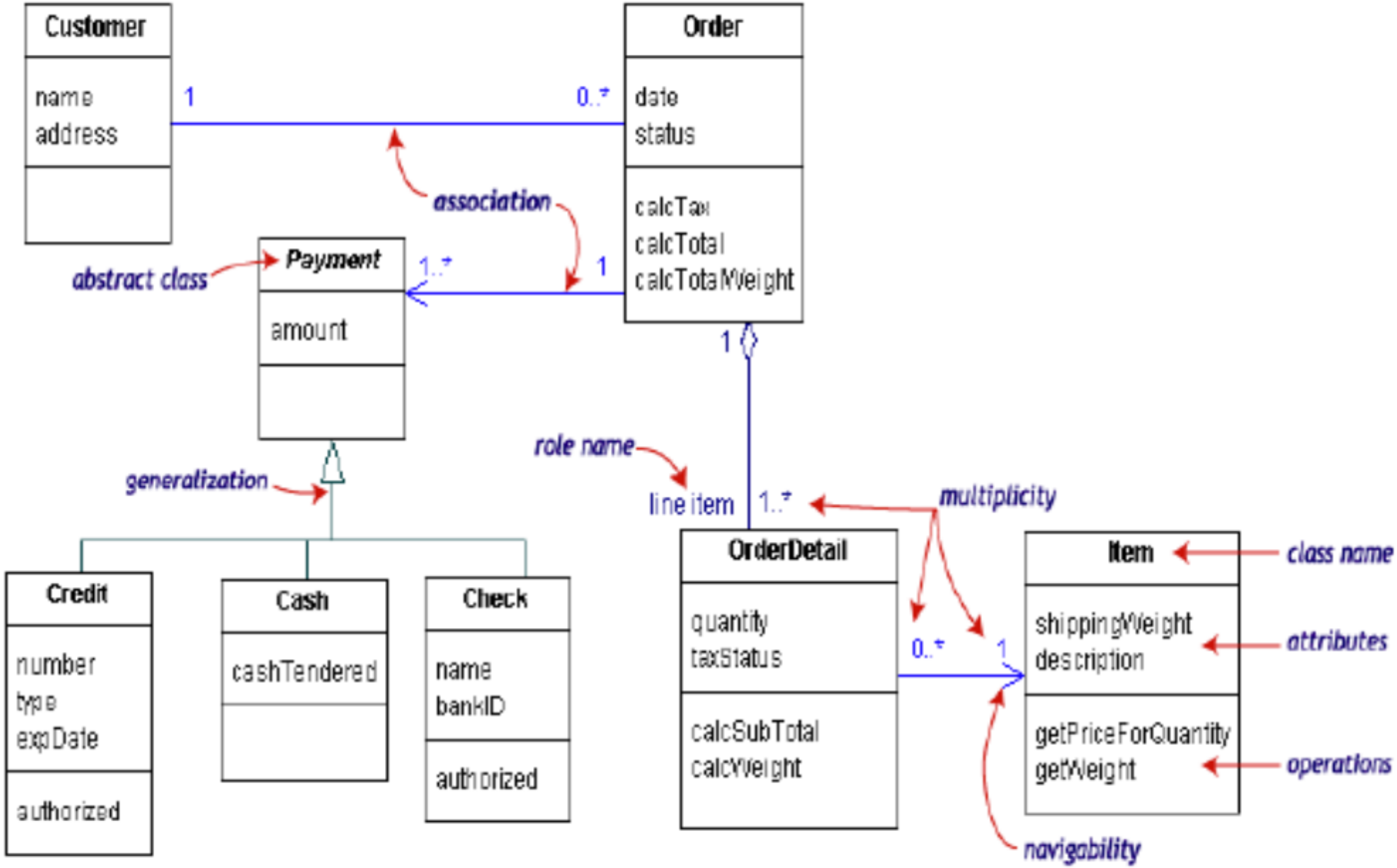
Simbol	Deskripsi
package 	*package* merupakan sebuah bungkusan dari satu atau lebih kelas
kelas 	kelas pada struktur sistem
antarmuka / *interface* 	sama dengan konsep *interface* dalam pemrograman berorientasi objek
asosiasi / *association* 	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
asosiasi berarah / *directed association* 	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*

SIMBOL CLASS DIAGRAM (2)

asosiasi berarah / *directed association* 	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*
generalisasi 	relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
kebergantungan / *dependency* 	relasi antar kelas dengan makna kebergantungan antar kelas
agregasi / *aggregation* 	relasi antar kelas dengan makna semua-bagian (*whole-part*)

CONTOH CLASS DIAGRAM

- Setiap diagram Class memiliki *Class* (kelas), *association*, dan *multiplicity*. Sedangkan *navigability* (alur arah) dan *role* (kegiatan) merupakan **optional** (tidak diharuskan).


Customer
name
address
1
0..*
Order
date
status
calcTax
calcTotal
calcTotalWeight
association
abstract class
Payment
amount
1..*
1
1
role name
line item
1..*
multiplicity
generalization
Credit
number
type
expDate
authorized
Cash
cashTendered
Check
name
bankID
authorized
OrderDetail
quantity
taxStatus
calcSubTotal
calcWeight
0..*
1
Item
shippingWeight
description
getPriceForQuantity
getWeight
class name
attributes
operations
navigability

CONTOH CLASS DIAGRAM (KASUS 1)

- Dalam Class Diagram Bagian Gudang
- Pertama supplier melakukan pemesanan barang melalui order,
- Kemudian Bagian Gudang akan mencatat pemesanan dan mengecek apakah stok barang masih ada atau sudah habis.
- Setelah itu Bagian Gudang akan mencatat bahan baku yang dibutuhkan kemudian membuat surat tagihan barang yang akan ditujukan kepada Bagian Pemesanan dan diberikan kepada customer.

CONTOH CLASS DIAGRAM (KASUS 1)