



# Pemodelan Data – Class Diagram -Desain dan Analisis Sistem -

---

NISA'UL HAFIDHOH, M.T.

PRODI DIII – TEKNOLOGI INFORMASI

# UML Diagrams (1)

○ Use Case Diagrams

Nodes:	Actor, Use (case)
Links:	Involvement, Extension, Usage

○ Class Diagrams

Nodes:	Class
Links:	Association, Generalization

○ Interaction Diagrams

Nodes:	Object
Links:	Message, Lifeline

○ State Diagrams

Nodes:	State, Sub-State
Links:	Transition

○ Activity Diagrams

Nodes:	Activity
Links:	Guard, Synchronization

○ Package Diagrams

Nodes:	Package
Links:	Dependency

○ Deployment Diagrams

Nodes:	Processor, Node
Links:	Dependency

# UML

---

## Use case Diagrams

- Menggambarkan perilaku fungsional sistem seperti yang terlihat oleh pengguna.

## Class diagrams

- Menggambarkan struktur statis sistem ini: [Classes](#), [Associations](#)

## Sequence diagrams

- Menggambarkan perilaku dinamis sebuah sistem: [Actors](#), [objects](#), [messages](#)

## Statechart diagrams

- Menggambarkan perilaku dinamis dari objek individu dari sistem: [states](#), [events](#), [transitions](#)

## Activity Diagrams

- Memodelkan perilaku dinamis sistem ini: [activities](#), [workflows \(flowcharts\)](#)

# Class Diagram

---

Class diagram adalah model statis yang menggambarkan struktur dan deskripsi class serta hubungannya antara class.

Pemahaman konseptual sistem terkait :

- Class apa saja yang dibutuhkan?
- Informasi dan fungsionalitas apa yang dimiliki class?
- Bagaimana class saling berinteraksi?
- Siapa yang bisa mengakses class?

# Class

---

Class adalah sebuah spesifikasi yang jika diinstansiasi 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 (metode/fungsi).

Contoh : Class Pintu

- Struktural / atribut : lebar, tinggi, warna
- Behavioral / metode : buka, tutup




# Representasi Class

---

Class direpresentasikan sebagai box dengan 3 bagian

- Nama class (dan stereotype)
- Atribut : ciri, karakteristik  
nama atribut : tipe
- Metode : fungsionalitas, apa yang bisa dilakukan  
nama metode(parameter : tipe)  
nama metode()

<b>BankAccount</b>	
owner : String balance : Double = 0.0	
deposit ( amount : Double ) withdraw ( amount : Double )	

# Visibilitas

---

Atribut dan metode (operation) dapat memiliki salah satu sifat berikut :

1. Private, hanya bisa dipanggil dari dalam kelas itu sendiri. method/atribut diawali “-”.
2. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan class turunannya. methode diawali dg tanda “#”.
3. Public, dapat dipanggil dari semua objek. methode/atribut diawali tanda “+”

BankAccount
-owner : String -balance : Double = 0.0
+deposit ( amount : Double ) -withdraw ( amount : Double)

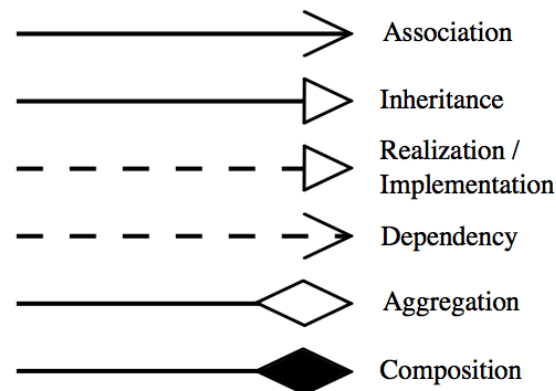
# Hubungan antar Class

---

Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah query antar class.

Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).

Pewarisan, yaitu hubungan hirarkis antar class. Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.



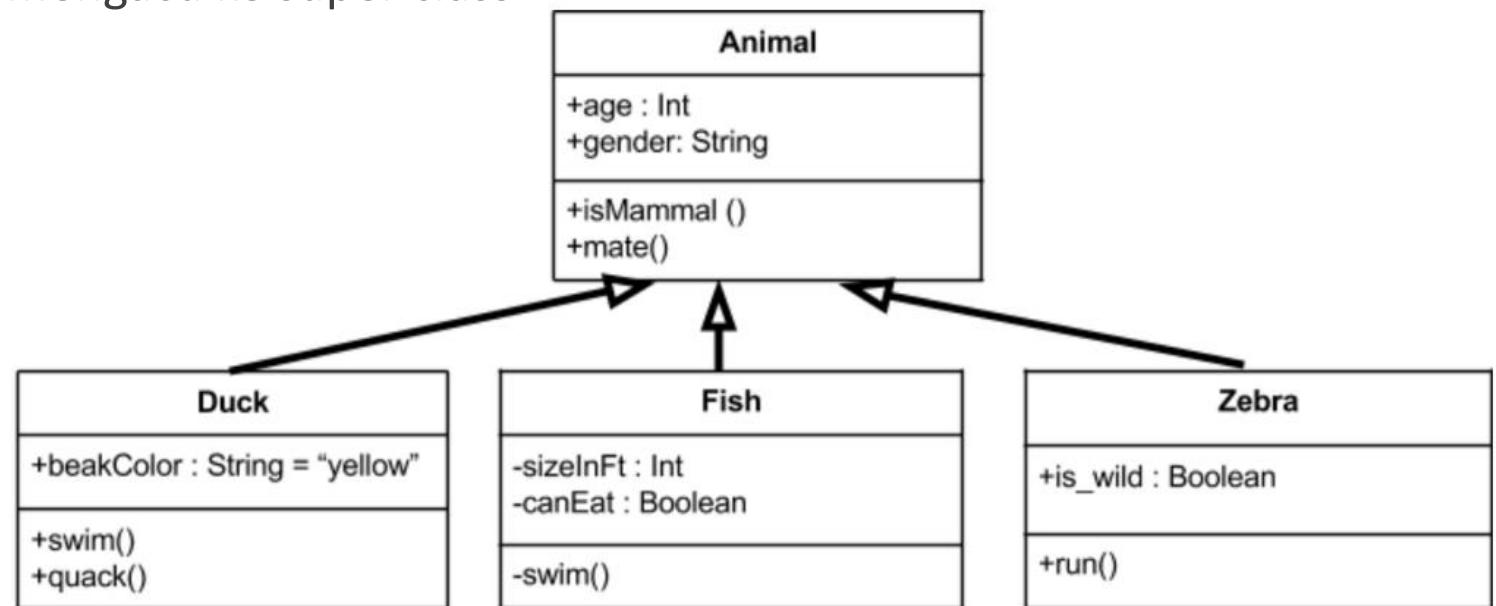


# Inheritance / Pewarisan

Menggambarkan sub class yang merupakan bentuk spesialisasi dari super classs.

Sub class mewarisi super classnya. Sub class mampu overriding metode super classnya

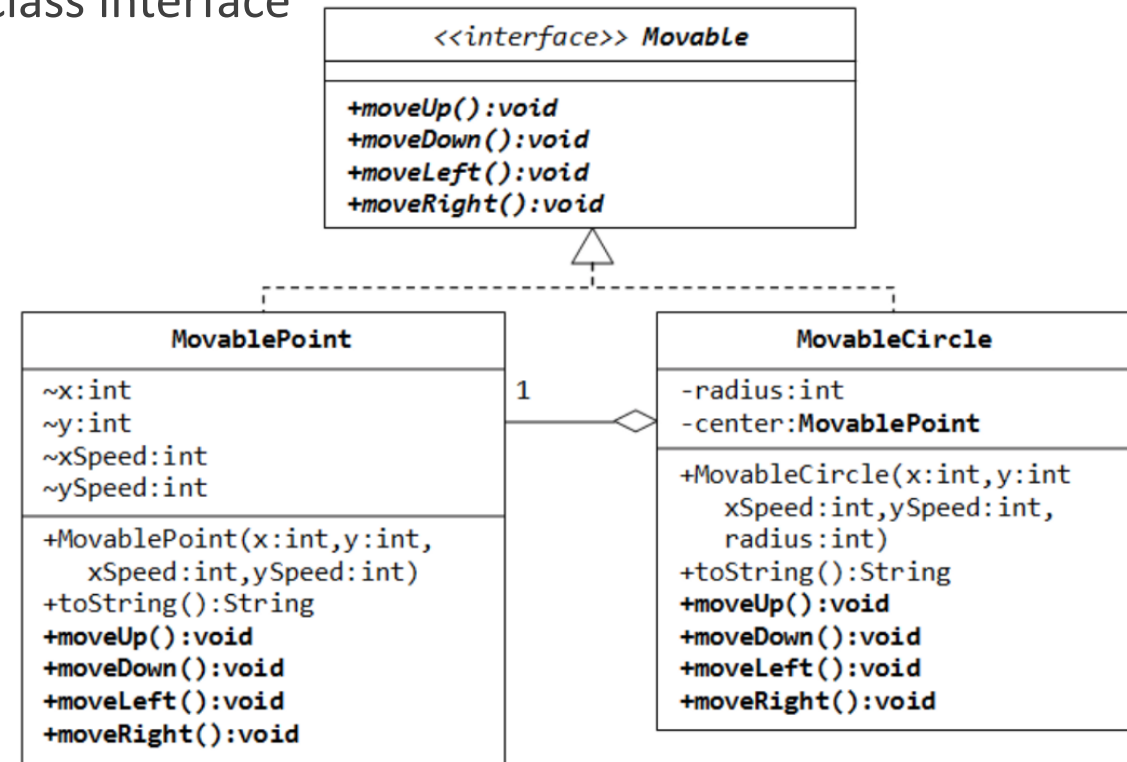
Dinotasikan dengan anak panah mengacu ke super class



# Interface

Hubungan antar class yang mengimplementasi / mengeksekusi method dari class lain.

Dinotasikan dengan garis putus dan anak panah ke class interface



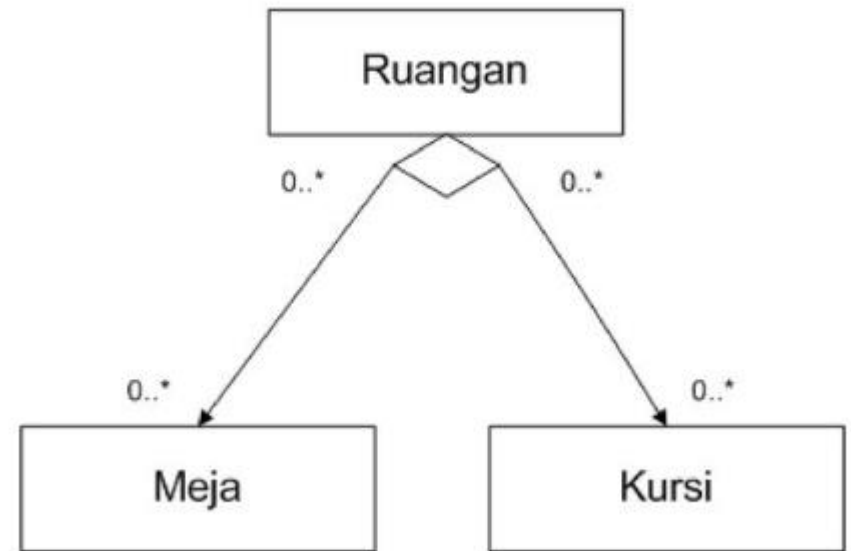
# Agregasi

Relasi satu kelas “whole” yang lebih besar berisi satu atau lebih kelas “part” yang lebih kecil. atau kelas “part” yang lebih kecil adalah bagian dari kelas “whole” yang lebih besar.

'has a' relationship

Contoh :

- Sebuah ruangan memiliki meja dan kursi
- Tanpa adanya ruang, meja dan kursi bisa tetap ada



# Komposisi

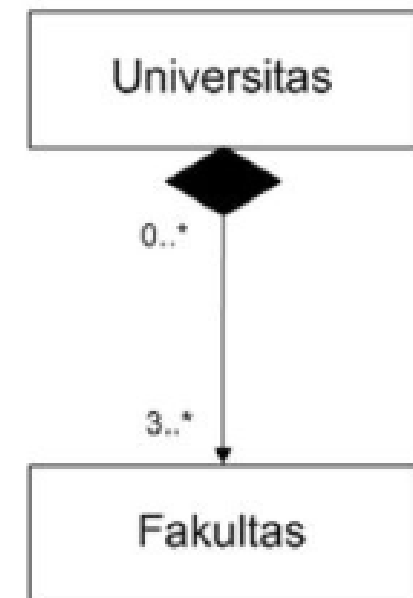
---

Bentuk terbatas dari agregasi yang menunjukkan ketergantungan antar class.

‘is part of’ relationship

Contoh :

- Kampus memiliki fakultas atau kampus terdiri dari fakultas (salah satunya)
- Tanpa ada kampus, maka tidak ada fakultas CS



# Stereotype Class

---

Entity Class : objects menyimpan seluruh data pada application

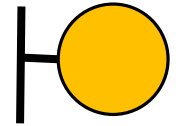
Boundary Class : digunakan untuk berinteraksi dengan sistem

- contoh : form, menus, dialog boxes

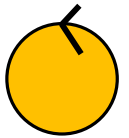
Control Class : mengatur seluruh kegiatan dan menangkap  
main application logic



Pesanan



FormPemesanan



PemrosesanPesanan

# Studi Kasus

---

Bank PNM memiliki beberapa teller yang bertanggung jawab terhadap transaksi nasabah. Nasabah yang memiliki rekening dapat melakukan setoran, penarikan serta mengajukan pinjaman.

Buatlah class diagram!

# Contoh Class Diagram

