Unified Modelling Language (UML)

Oleh: Rahmi Hidayati, S.Kom., M.Cs.

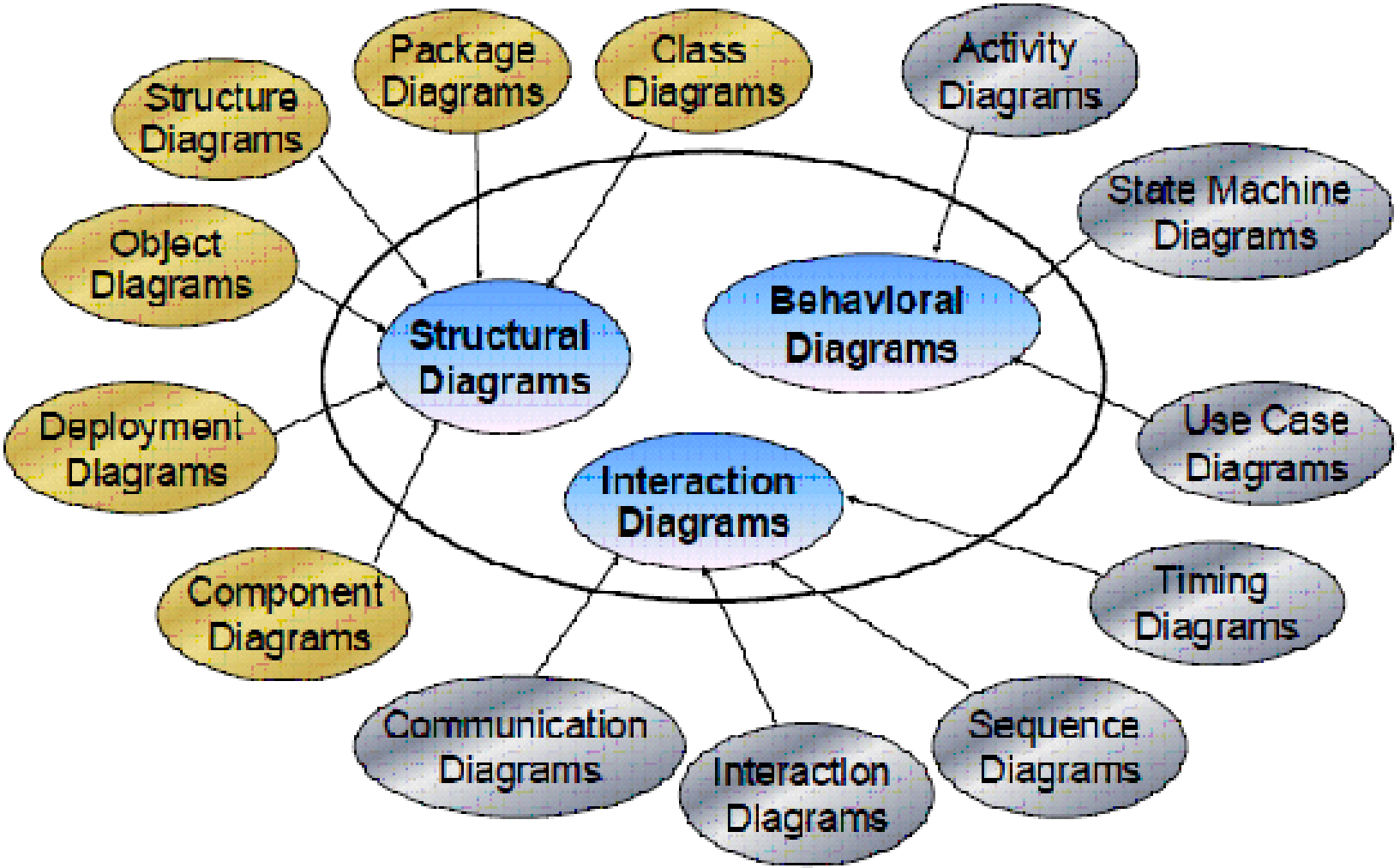
Tujuan Pembelajaran

- Mahasiswa mampu menjelaskan diagram UML
- Mahasiswa mampu memahami penggunaan use case, activity diagram, sequence diagram dan class diagram

Unified Modelling Language (UML)

- Aturan-aturan pemodelan yang digunakan untuk mendeskripsikan sistem perangkat lunak dalam bentuk kumpulan obyek.
- UML bukan sebuah metode untuk mengembangkan sistem, tetapi notasi-notasi yang digunakan secara umum sebagai standar untuk pemodelan obyek.
- UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak .

UML 2.0 Diagrams

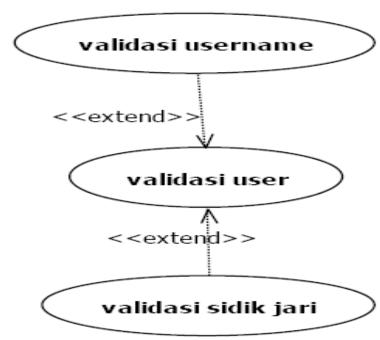


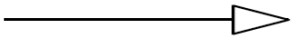
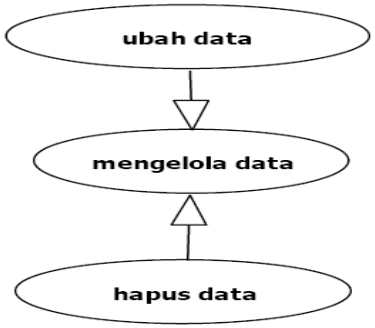
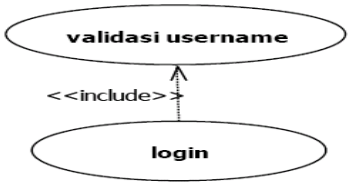
Use Case

- Pemodelan untuk menggambarkan kelakuan(*behavior*) sistem yang akan dibuat.
- *Diagram use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat.
- Terdapat beberapa simbol dalam menggambarkan diagram *use case*, yaitu *use cases*, aktor dan relasi.

Simbol-Simbol *Use Case*

Simbol	Deskripsi
Use case 	fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case*
Aktor / *actor*	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang;

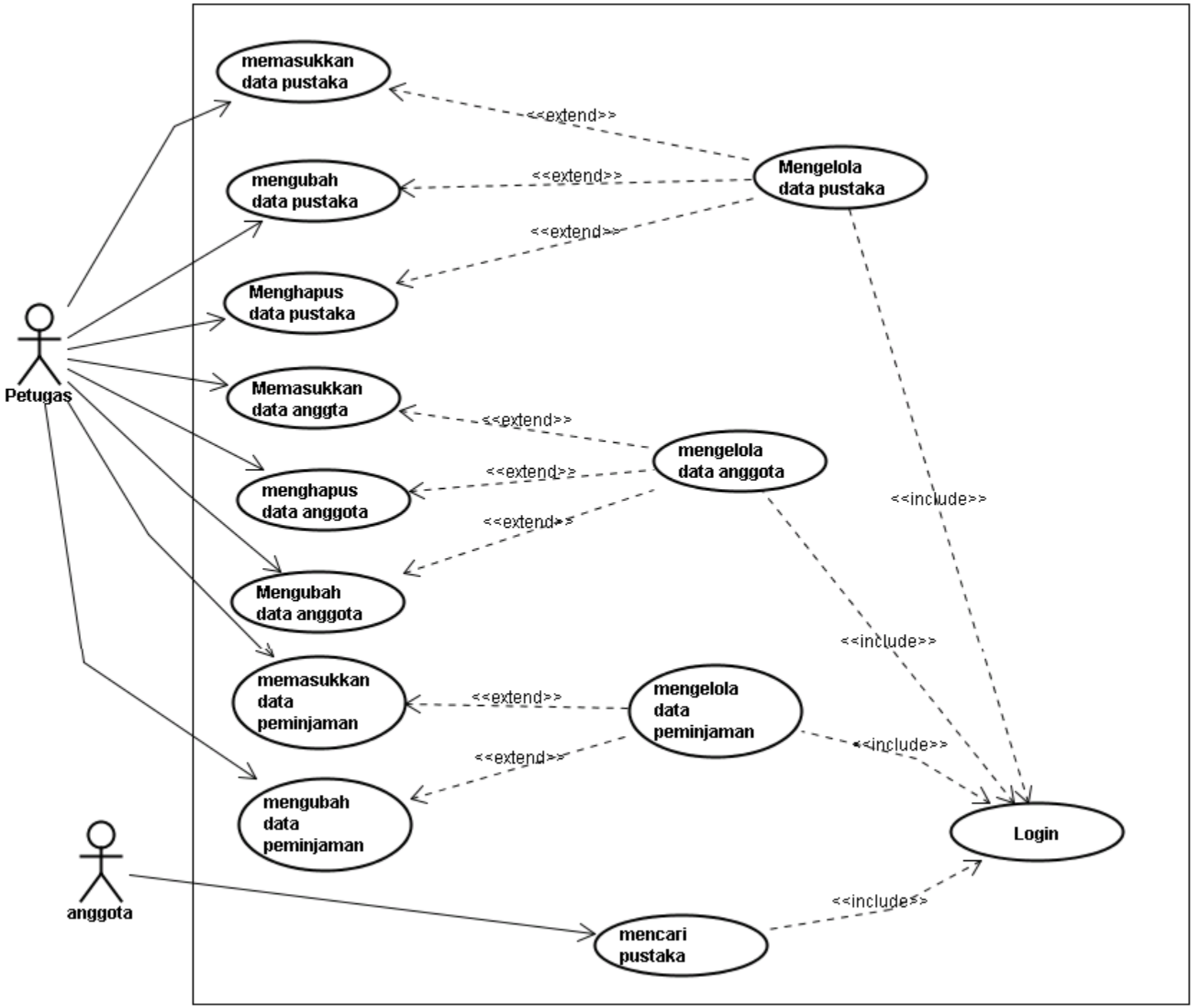
Simbol	Deskripsi
	biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Asosiasi / *association*	komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor
Ekstensi / *extend* 	relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan, misal  arah panah mengarah pada *use case* yang ditambahkan

Simbol	Deskripsi
	dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:  arah panah mengarah pada *use case* yang menjadi generalisasinya (umum)
Menggunakan / *include* / *uses*   	relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini ada dua sudut pandang yang cukup besar mengenai include di *use case*: • include berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahan dijalankan, misal pada kasus berikut: 

Contoh :

Pemecahan studi kasus untuk sistem informasi perpustakaan, langkah pertama yaitu melakukan pencarian aktor. Yaitu berkaitan dengan SIAPA, PERAN dan NILAI apa yang akan didapatkan.

No	Aktor	Deskripsi
1.	Petugas perpustakaan	orang yang bertugas dan memiliki hak akses untuk melakukan operasi pengelolaan data pustaka, anggota, dan proses pemiinjaman pustaka
2.	Anggota/pengunjung perpustakaan	anggota adalah orang yang diperbolehkan meminjam pustaka sesuai dengan hak aksesnya, sedangkan pengunjung hanya memiliki hak akses melihat pustaka dan membaca di perpustakaan tanpa memiliki hak untuk meminjam pustaka.


memasukkan data pustaka
mengubah data pustaka
Menghapus data pustaka
Memasukkan data anggta
menghapus data anggota
Mengubah data anggota
memasukkan data peminjaman
mengubah data peminjaman
Mengelola data pustaka
mengelola data anggota
mengelola data peminjaman
Login
mencari pustaka
Petugas
anggota
<<extend>>
<<extend>>
<<extend>>
<<extend>>
<<extend>>
<<extend>>
<<extend>>
<<extend>>
<<include>>
<<include>>
<<include>>
<<include>>

- Tahap selanjutnya adalah menemukan use case. berkaitan dengan INFORMASI apa yang akan diberikan oleh sistem kepada aktor.

No	*Use case*	Deskripsi
1.	Memasukkan data pustaka	merupakan proses memasukkan data pustaka ke dalam basis data
2.	Memasukkan data anggota	merupakan proses memasukkan data anggota ke dalam basis data
3.	Memasukkan data peminjaman	merupakan proses memasukkan data peminjaman ketika ada anggota yang meminjam pustaka
4.	Mencari pustaka	mencari pustaka berdasarkan judul, nama pengarang, jenis, dan kode pustaka dimana akan menampilkan data pustaka yang dicari

- Tahap ketiga adalah membuat skenario per-use case.

Memasukkan data pustaka

Aksi Aktor	Reaksi Sistem
1. Memasukkan data pustaka seperti judul buku, penerbit, tahun terbit, pengarang, jumlah halaman, kondisi buku di menu memasukkan pustaka	
2. Menekan tombol "Simpan"	
	3. Mengecek valid tidaknya data masukan
	4. Jika data pustaka yang dimasukkan valid, maka data pustaka akan disimpan di database dan akan menampilkan pesan "sukses disimpan"
Alur alternatif No 4 a. Jika data pustaka yang dimasukkan tidak valid, maka akan menampilkan pesan "tidak sukses disimpan"	

Alur alternatif untuk memperbaharui data pustaka	
Aksi Aktor	**Reaksi Sistem**
1. Memasukkan judul buku atau IDBuku	
2. Menekan tombol "Cari"	
	3. Menampilkan informasi buku yang terdiri dari judul buku, penerbit, tahun terbit, pengarang, jumlah halaman, kondisi buku, letak buku
4. Memperbaharui data. Beberapa pilihan data yang dapat diperbaharui diantaranya judul buku, penerbit, tahun terbit, pengarang, jumlah halaman, kondisi buku, letak buku	
5. Menekan tombol "Simpan"	
	6. Mengecek valid tidaknya data yang diperbaharui.
	7. Jika data yang dimasukkan valid, maka data pustaka yang baru akan disimpan di database dan menampilkan pesan "sukses disimpan"
Alur alternatif No 7 a. Jika data pustaka yang dimasukkan tidak valid, maka akan menampilkan pesan "tidak sukses disimpan"	

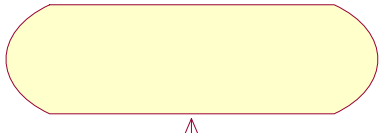
Alur alternatif untuk menghilangkan data pustaka	
Aksi Aktor	**Reaksi Sistem**
1. Memasukkan judul buku atau IDBuku	
2. Menekan tombol "Cari"	
	3. Menampilkan informasi buku yang terdiri dari judul buku, penerbit, tahun terbit, pengarang, jumlah halaman, kondisi buku, letak buku
4. Menekan tombol "Hapus"	
	5. Menampilkan pesan "Yakin akan dihapus?"
	6. Jika jawaban pesan adalah "Ya", maka data pustaka IDBuku yang dicari akan dihapus dari database.
Alur alternatif No 6 a. Jika jawab pesan adalah "Tidak", maka akan ditampilkan menu pustaka	

Activity Diagram

- ***Activity diagram*** : diagram yang digunakan untuk menggambarkan
  - Proses bisnis
  - Langkah-langkah use case
  - Logika perilaku obyek/ metode
- *Activity diagram* adalah cara lain menggambarkan *flow of events*.
- Menunjukkan kontrol aliran dari *activity* ke *activity*.

Activity

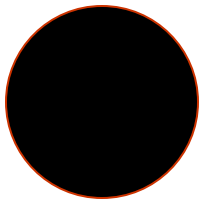
- Activity menggambarkan sebuah pekerjaan atau tugas dalam *workflow*.
- Pada UML, *activity* digambarkan dengan simbol berikut :



Activity

Start State

- *Start state* menunjukkan dimulainya suatu *workflow* pada sebuah *activity diagram*.
- Hanya ada satu *start state* dalam sebuah *workflow*.
- Pada UML, *start state* digambarkan dengan simbol lingkaran yang solid.

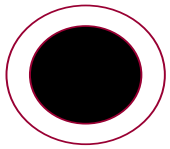


Start State

End State

- *End state* menggambarkan akhir atau terminal dari pada sebuah *activity diagram*.
- Bisa terdapat lebih dari satu *end state* pada sebuah *activity diagram*.
- Pada UML, *end state* digambarkan dengan simbol sebuah *bull's eye*.

End State



State Transitions

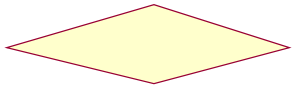
- *State transition* menunjukkan kegiatan apa berikutnya setelah suatu kegiatan sebelumnya.
- Pada UML, *state transition* digambarkan oleh sebuah *solid line* dengan panah.

State Transition

Decisions

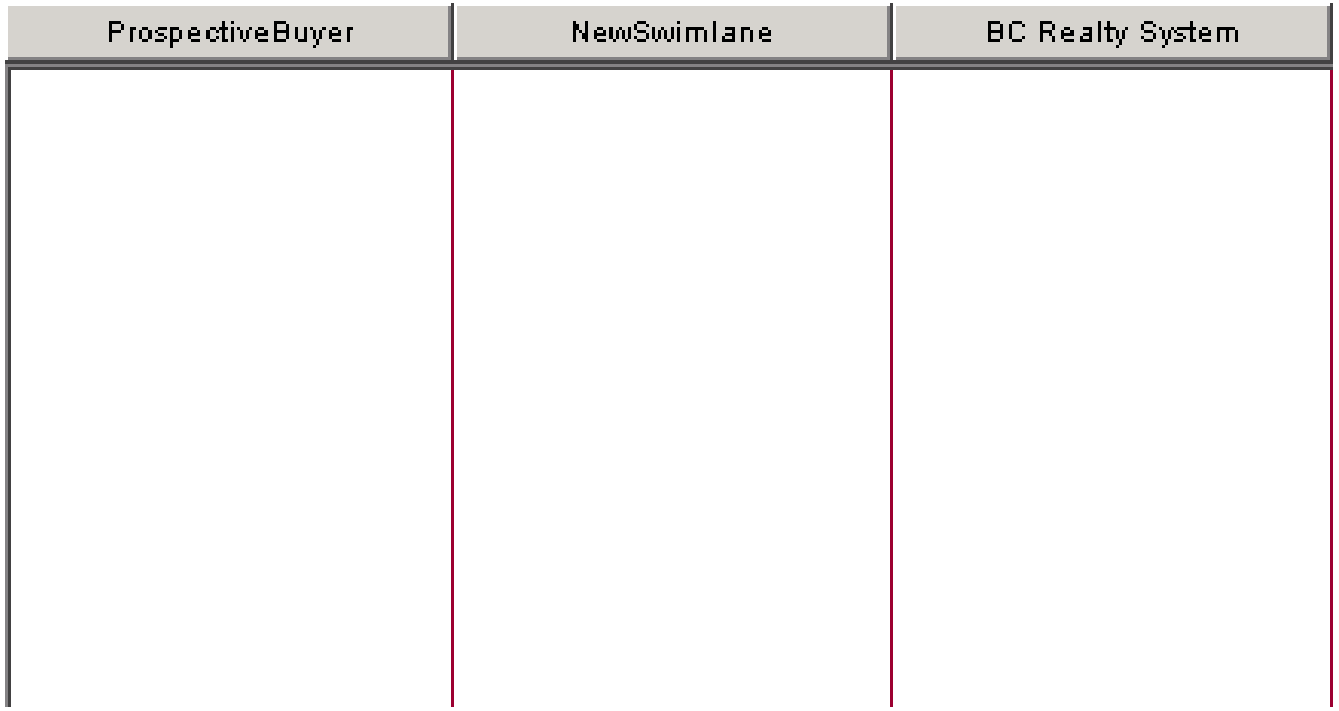
- *Decision* adalah suatu titik (*point*) pada *activity diagram* yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi.
- Pada UML, *decision* digambarkan dengan sebuah simbol diamond.

Decision

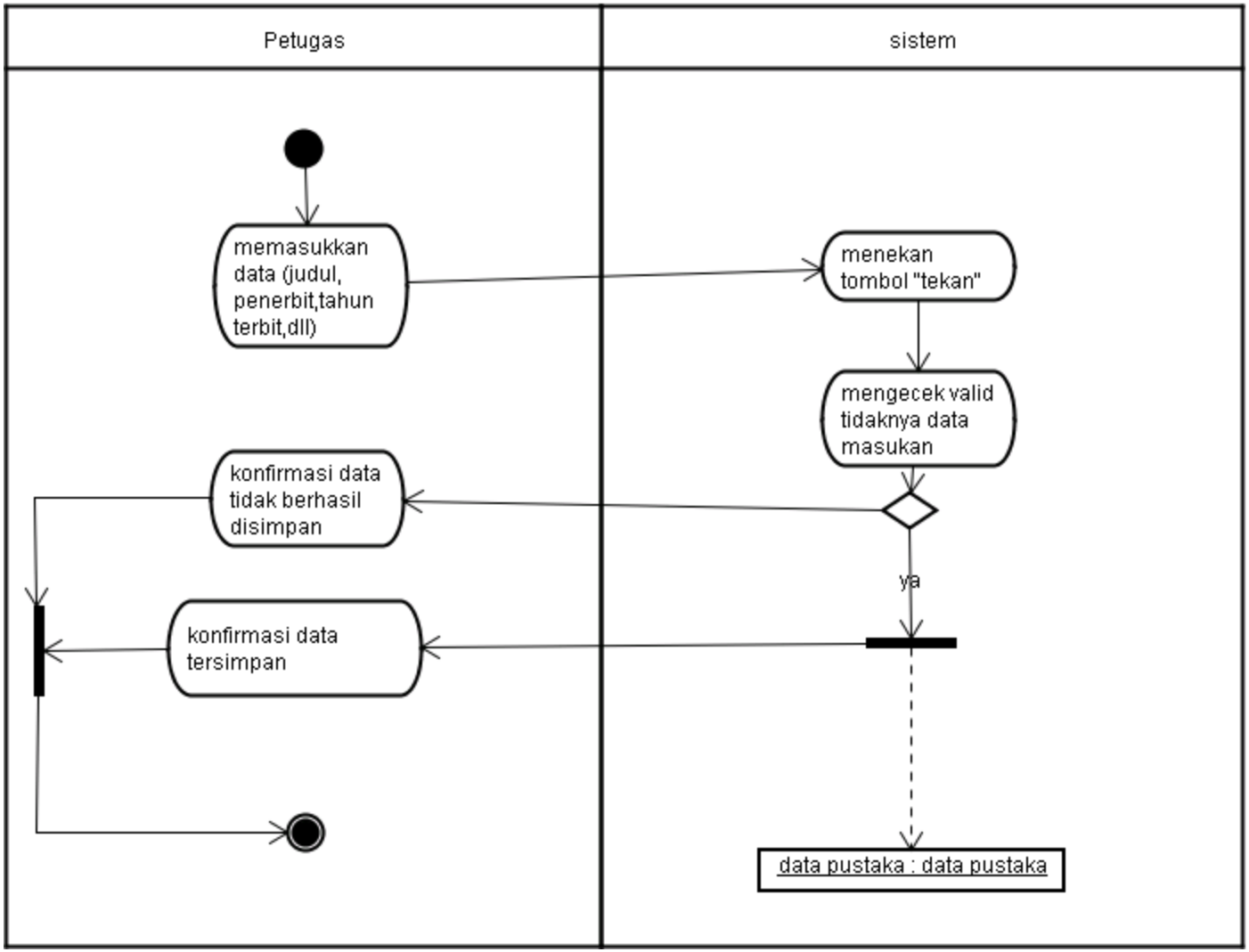


Swimlanes

- A swimlane is used to partition an activity diagram to help us better understand who or what is initiating the activity.



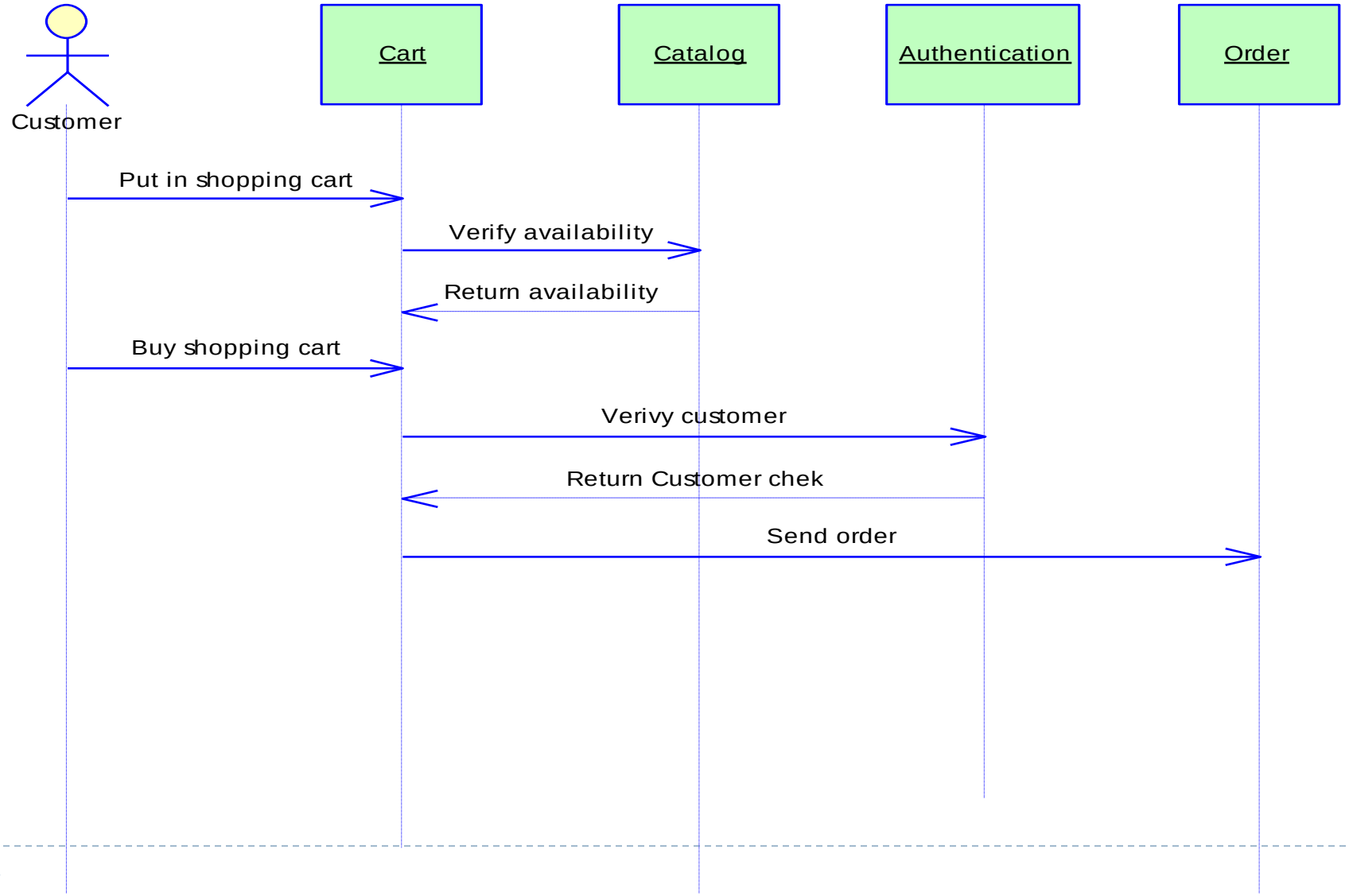
Activity Diagram



Sequence Diagram

- *Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display dan sebagainya). Berupa *message* yang digambarkan terhadap waktu.
- *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).
- *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu.

Proses pemesanan buku



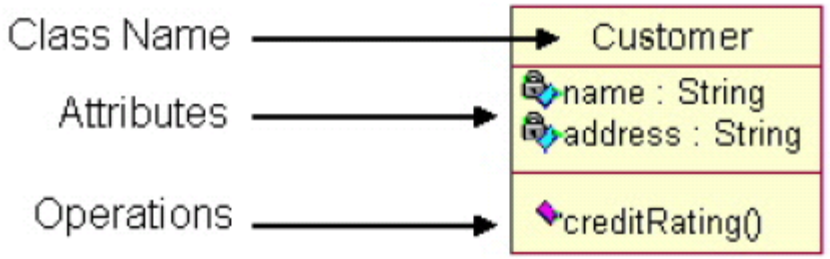
Class Diagram

- *Class diagram* atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.
  - Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
  - Atribut mendeskripsikan properti dengan sebaris teks di dalam kotak kelas tersebut.
  - Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.
- Kelas memiliki tiga area pokok :

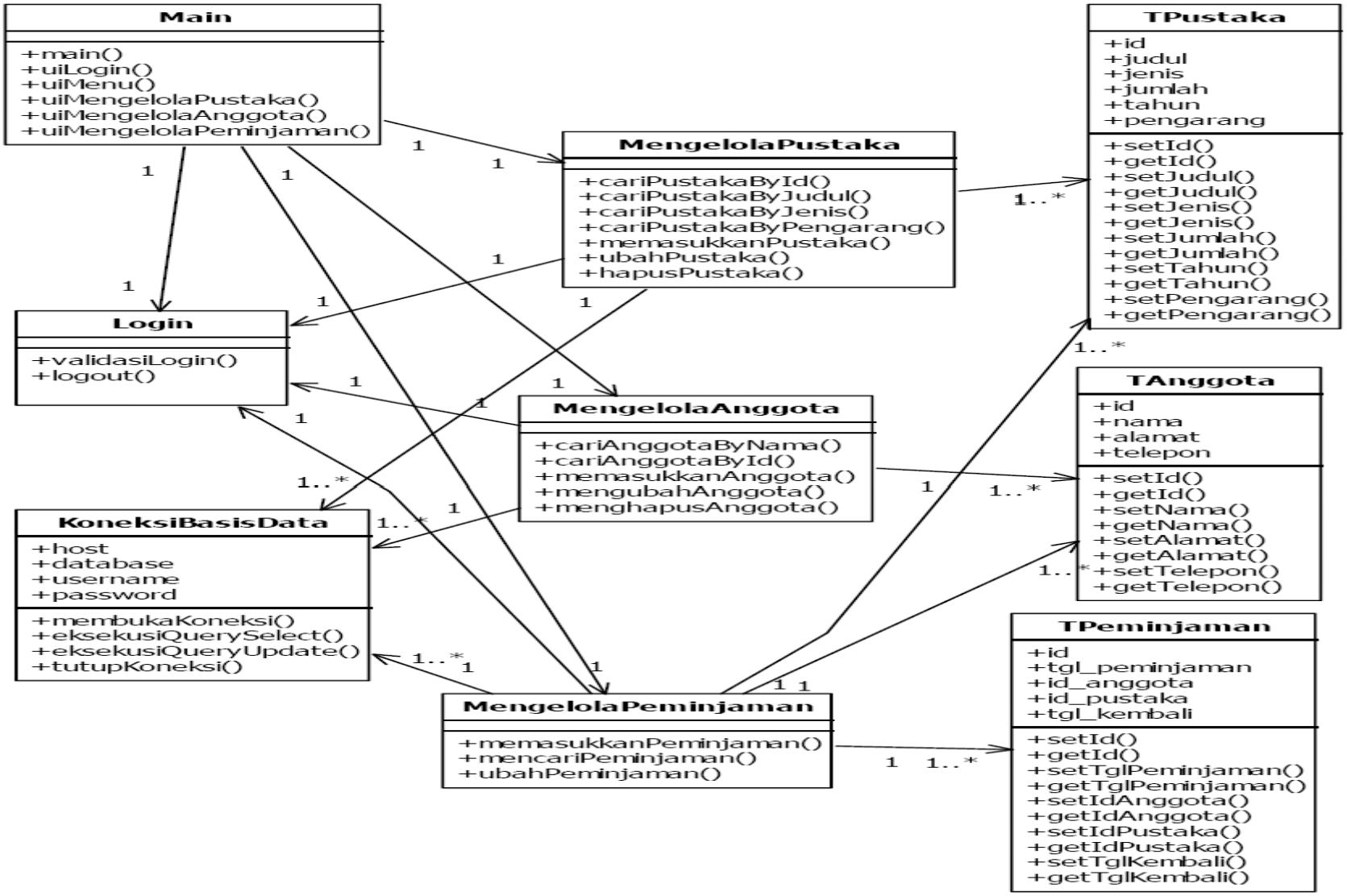
  1. Nama
  2. Atribut
  3. Operasi

Contoh kelas : Customer

- Atribut: name, address
- Method/Operasi: creditRating



Contoh Diagram Kelas untuk SI Perpustakaan

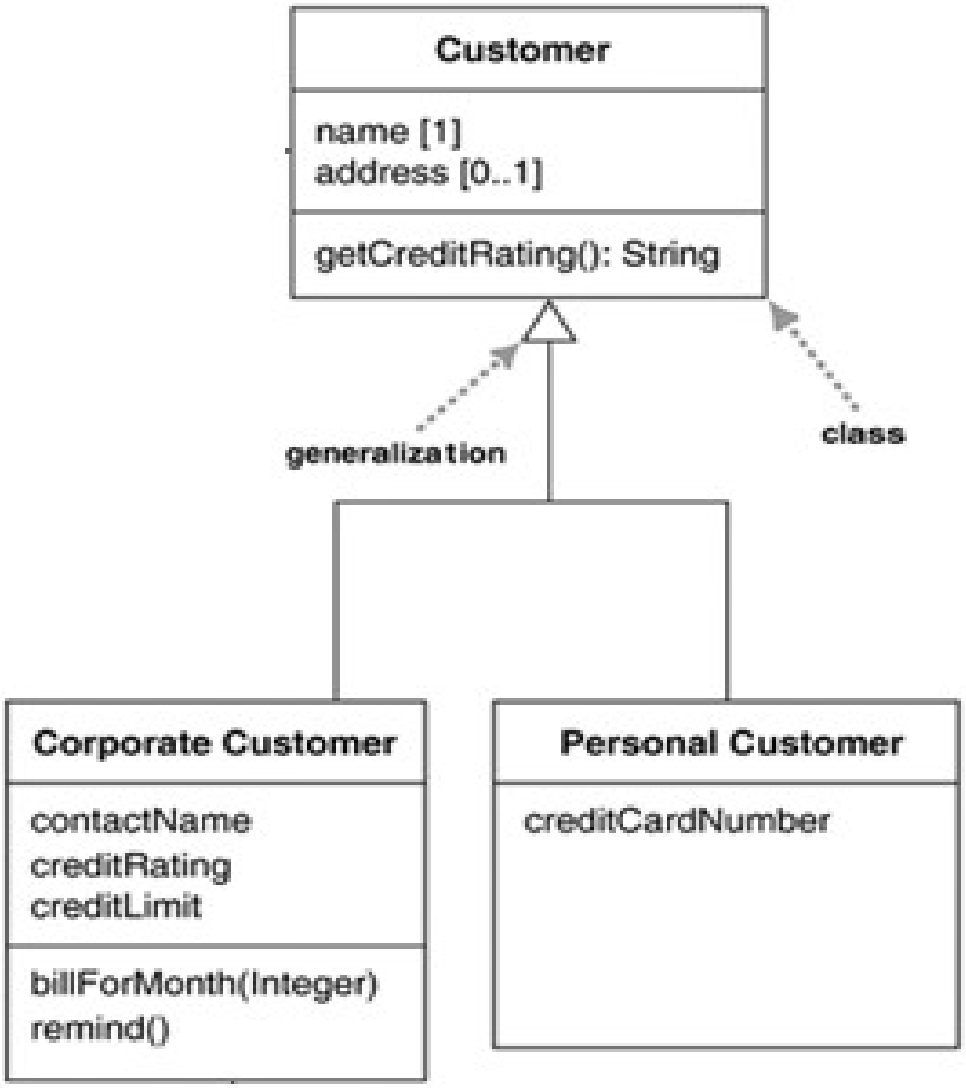


Multiplicity

- Indikasi berapa banyak objek yang bisa mengisi properti :
  - 1 (pasti 1)
  - 0..1 (0 atau 1)
  - * (Tidak ada batasan, bisa 0, 1, ..., n)
- Biasanya didefinisikan batas bawah dan atas, kecuali untuk yang pasti bernilai 1.
- Mirip dengan konsep *one-to-one* dan *one-to-many* pada *relational database*.

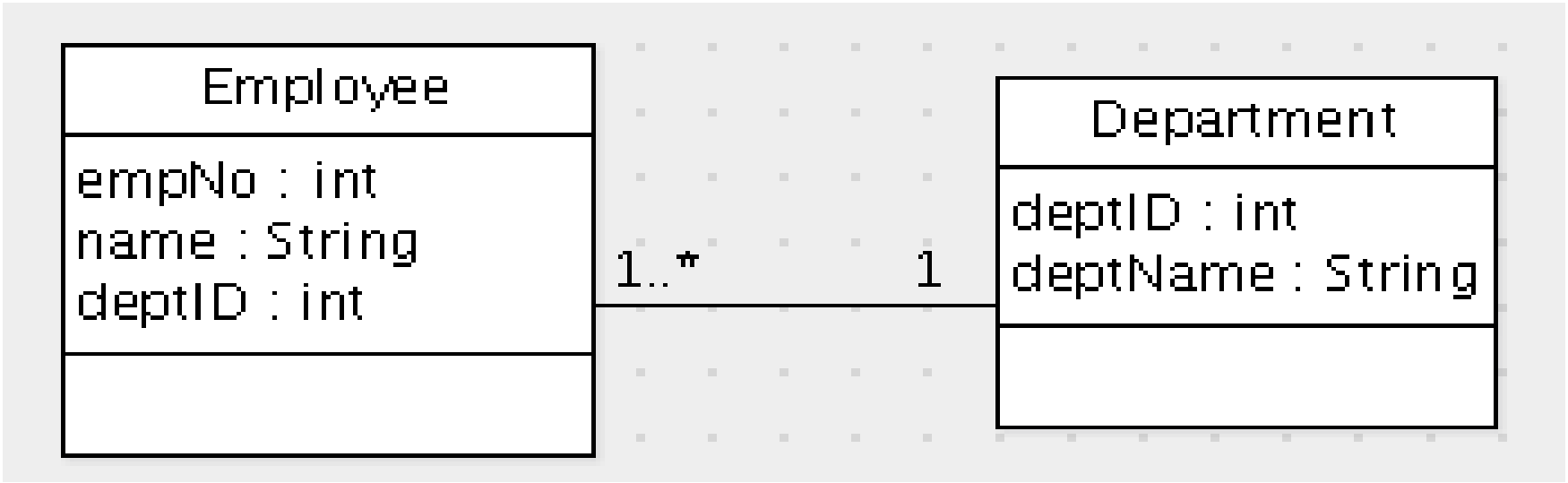
Generalization

- *Inheritance* pada UML.
- *Sub class* mewarisi *feature* dari *super class*-nya.
- Dinotasikan dengan anak panah mengacu ke *super class*.



Associations

- Menggambarkan hubungan antar class.
- Ditandai dengan garis lurus.
- Seringkali ditambahkan label dan *multiplicity* untuk memperjelas hubungan



Aggregation

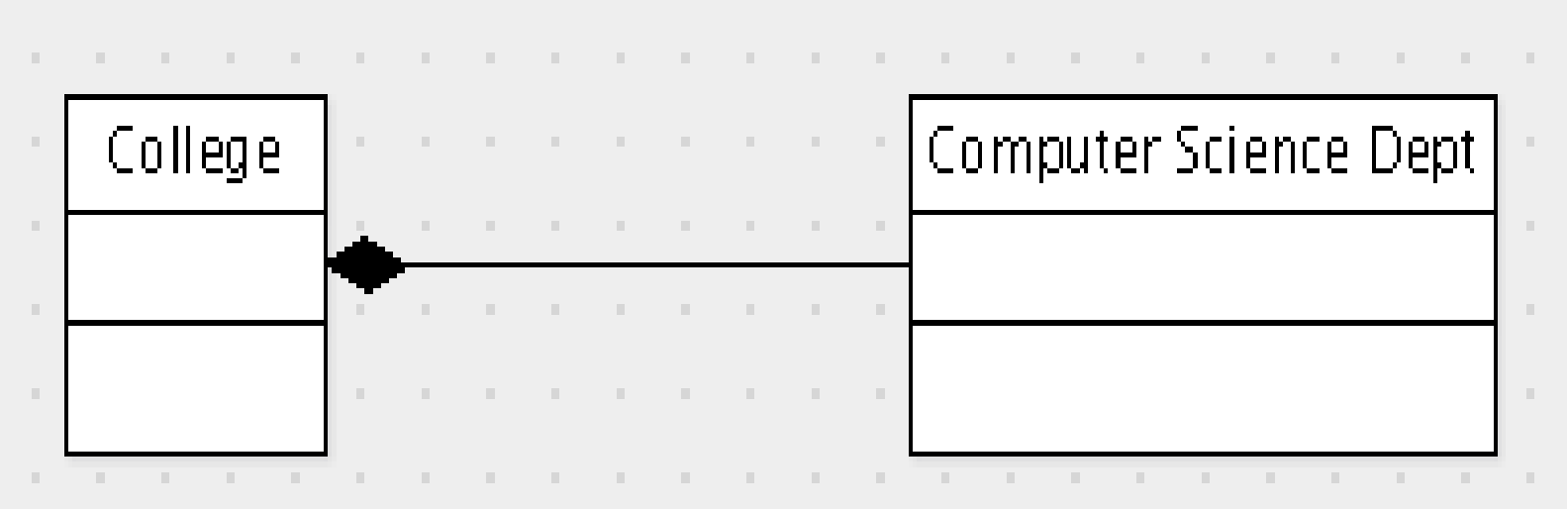
- Relasi biasa disebut 'has a' *relationship*
  - Klub memiliki banyak anggota
  - Orang bisa memiliki makna tersendiri tanpa kehadiran sebuah klub
- Dinotasikan dengan *diamond* "kosong".
- Jika dipisah, tidak merubah makna.

Aggregation

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

Composition

- 'has a' or 'contains a' relationship (*whole-part*).
- Kampus memiliki fakultas CS atau kampus terdiri dari fakultas CS (salah satunya).
- Tanpa ada kampus, maka tidak ada fakultas CS



Penggunaan UML

- Pada tahap proposal untuk analisa desain sistem cukup berupa:
  - *Diagram Use case*
  - *Activity Diagram*
- Pada tahap laporan TA, berupa:
  - *Diagram Use case*
  - *Activity Diagram*
  - *Diagram Sequence*
  - *Class Diagram*

Tool Yang Mendukung UML

- Rational Rose (www.rational.com)
- Together (www.togethersoft.com)
- Object Domain (www.objectdomain.com)
- Jvision (www.object-insight.com)
- Objecteering (www.objecteering.com)
- MagicDraw (www.nomagic.com/magicdrawuml)
- Visual Object Modeller (www.visualobject.com)
- Edraw UML Diagram

Latihan

- Sebutkan bagian dari structural diagram pada diagram UML.
- Jelaskan perbedaan antara include dan extend pada use case diagram.
- Jelaskan fungsi dari activity diagram dan sequence diagram.
- Sebutkan bagian pada class diagram beserta contoh penggunaannya.
- Jelaskan perbedaan penggunaan asosiasi dan generalisasi.

TERIMA KASIH