



SCENARIO BASED MODELING REVIEW

IF3250 PROYEK PERANGKAT LUNAK

Sem II 2024/2025

Scenario-based Modeling

- Apa yang dimodelkan?
 - Cara pengguna **berinteraksi** dengan perangkat lunak
 - Cara perangkat lunak **berinteraksi** dengan sistem lain
- Seperti apa modelnya?
 - Diagram *use-case*
 - Diagram aktivitas (dengan dan tanpa *swimlane*)
 - Diagram interaksi (mis. *diagram sequence*)

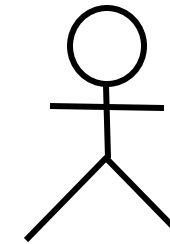
Diagram *Use-case*

- *Use-case*
- Actor
- Scenario

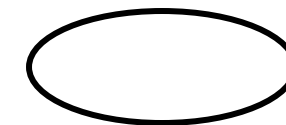
Ivar Jacobson: “[**Use-cases**] are simply an aid to defining what exists outside the system (**actors**) and what should be performed by the system (**use-cases**).”

Konsep Pemodelan *use-case*

- Aktor mewakili semua yang **berinteraksi** dengan sistem
 - Aktor adalah unsur '**eksternal**'
- *Use-case* adalah **urutan aksi-aksi** dalam sistem yang melakukan suatu pekerjaan yang memberikan suatu hasil untuk aktor
- Use-case bertindak sebagai **penghubung** antara pengguna dengan pengembang
- Use-case berguna sebagai **alat komunikasi** antara pengguna dan pengembang



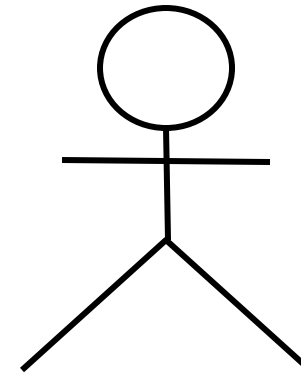
Actor



Use-Case

Actor

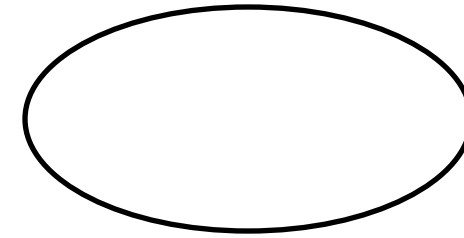
- Aktor bukan bagian dari sistem
- Aktor mungkin secara aktif bertukar informasi dengan sistem
- Aktor mungkin berfungsi pasif sebagai penerima informasi
- Aktor bisa merepresentasikan
 - Manusia,
 - Mesin,
 - Sistem lain



Actor

Aktor mewakili unsur Eksternal

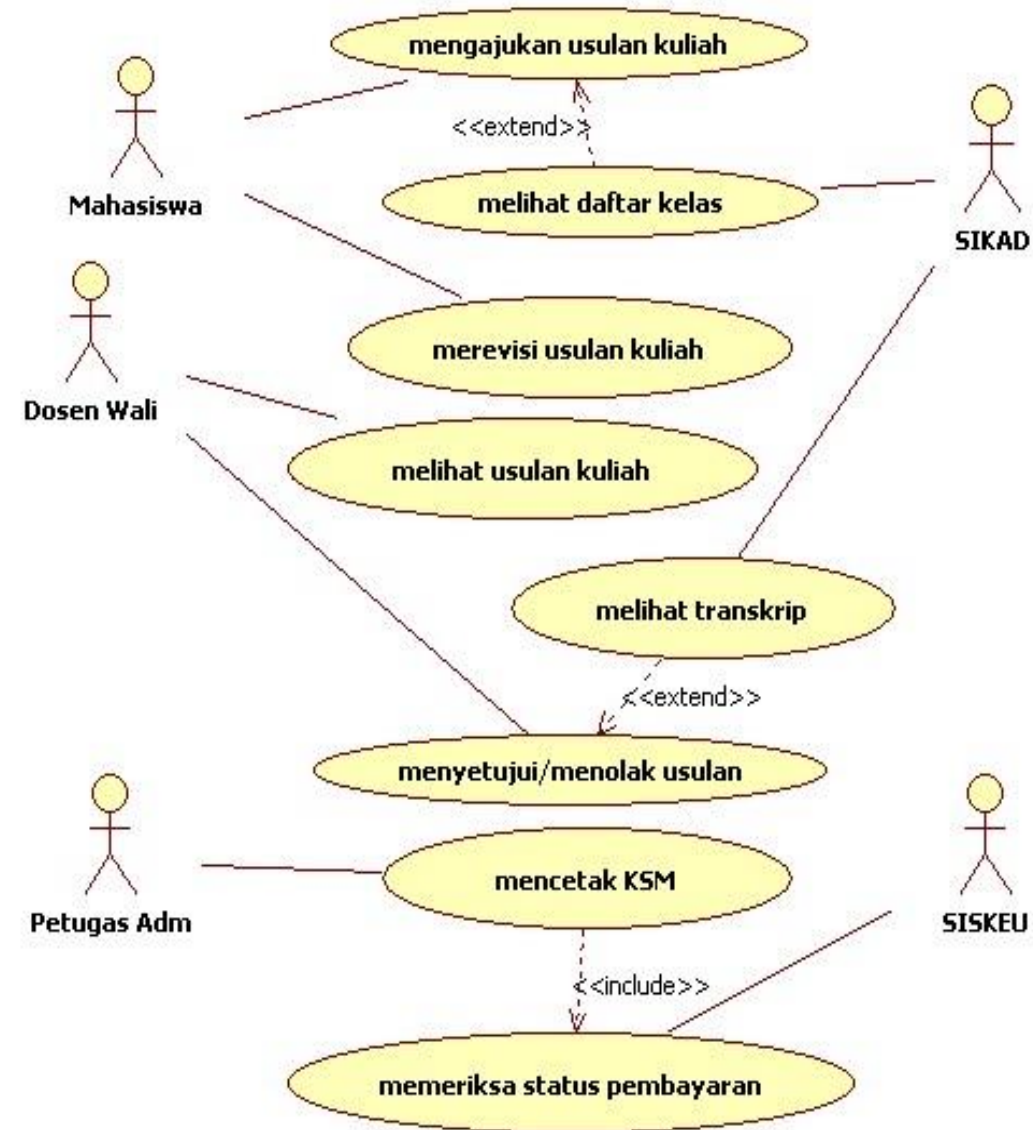
Use-Case



Use-Case

- *Use-case* merepresentasikan **dialog** antara aktor dengan sistem
 - *use-case* di'**inisiasi**' oleh **aktor** untuk melakukan suatu fungsi tertentu dalam sistem
- *Use-case* merepresentasikan dialog antara satu atau lebih aktor
 - sistem akan **mengembalikan** suatu **nilai** ke aktor
- *Use-case* perlu menggambarkan **event** yang lengkap dan memiliki makna bagi sistem
- *Use case* juga dapat dilihat sebagai tujuan sistem secara umum yang mungkin melibatkan satu atau lebih aktor
- *Use-case* mengarahkan ke penggunaan sistem

Contoh Use-case Diagram



Penjelasan *Use-case*

- Gambar *use-case* (termasuk aktornya) perlu disertai dengan **keterangan** yang akan membantu **menjelaskan** gambaran yang diberikan
 - Tidak perlu panjang lebar, untuk setiap *use-case* lebih kurang **dua baris** saja.
- Perlu dijelaskan bagaimana antar *use-case* **saling terkait**
 - Bagaimana *use case* itu dilakukan oleh actor
 - Bagaimana suatu *use case* terlibat dengan *use-case* lain

→ Deskripsi Use Case

Deskripsi Use Case - contoh

Use Case: Mengajukan Usulan Kuliah
Iteration : ke-2,
Modifikasi terakhir 1 Maret 2018
Primary Actor: Mahasiswa
Goal in Context: Untuk pengajuan usulan
kuliah oleh mahasiswa
Preconditions: Mahasiswa sudah terdaftar
dan mahasiswa sudah memasukkan
nama user
dan password sebelumnya
Trigger: Jika Mahasiswa memutuskan untuk
mengambil kuliah di awal semester
Scenario:
1. Mahasiswa memilih menu entri usulan kuliah
2. Sistem menampilkan form entri FRS
3. Mahasiswa mengisikan kode kuliah
4. Sistem menampilkan informasi detail matakuliah
(nama, sks)
5. Mahasiswa menekan tombol SIMPAN
6. Sistem menyimpan data usulan ke dalam
basisdata

Exception:

1. Mahasiswa memilih menu entri usulan kuliah
2. Mahasiswa memilih untuk melihat daftar kelas
3. Sistem menampilkan daftar kelas yang dibuka
4. Mahasiswa memilih matakuliah dari daftar
5. Mahasiswa menekan tombol SIMPAN
6. Sistem menyimpan data usulan ke dalam
basisdata

Priority: Prioritas sedang

When available: Iterasi ketiga

Secondary Actor: tidak ada

Open Issues:

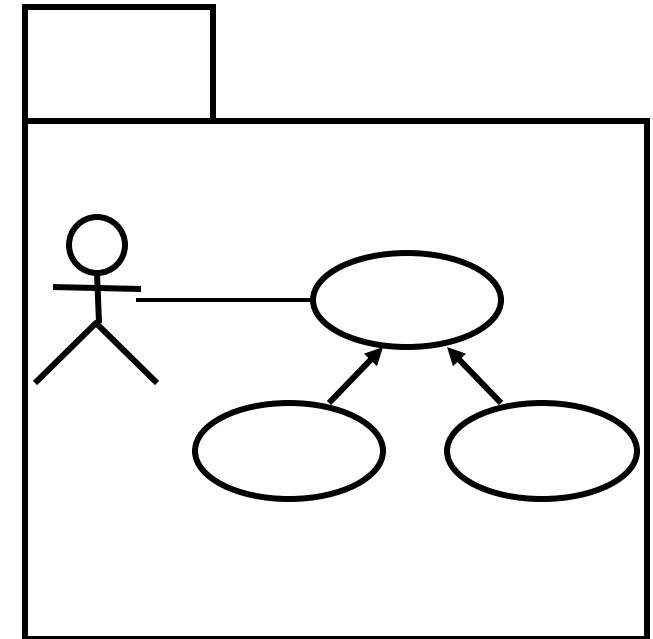
1. Bagaimana mekanisme mendeteksi, jika ada suatu
kuliah memiliki prerequisite kuliah lain
2. Bagaimana jika mahasiswa ingin membatalkan usulan

Deskripsi Use Case

- Untuk jumlah *use-case* yang **besar** dengan berbagai **alternatif**, maka penulisan teks menjadi tidak praktis, jadi kadang digunakan **diagram**
 - **Diagram Statechart**
 - Berisi penjelasan *state* dan transisi dalam *use-case*
 - **Diagram Aktivitas**
 - Menggambarkan urutan aksi
 - **Diagram Interaksi**
 - Menjelaskan interaksi antar instansiasi dari aktor dan instansiasi dari *use-case*

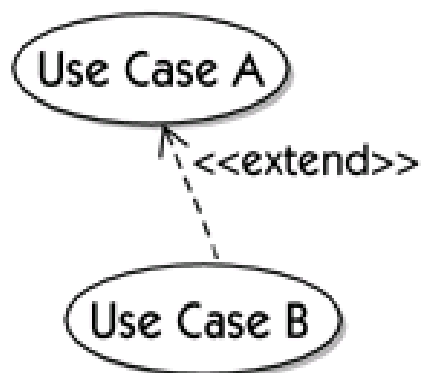
Paket (Packages) dalam model Use-case

- **Paket** digunakan untuk **mengelompokkan** elemen-elemen yang terkait secara **semantik**
- Kegunaan:
 - *Use-case* lebih terstruktur
 - **Batasan** lingkup dari satu atau beberapa *use-case*
 - Paket dalam *use-case* juga bisa digunakan untuk
 - Menunjukkan urutan sistem
 - Konfigurasi sistem
 - *Delivery unit*
 - Memudahkan pembagian pekerjaan dalam tim

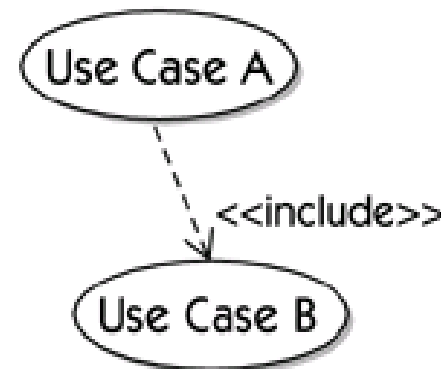


Ketergantungan antar use-case

- Hubungan **extend** menunjukkan kemungkinan adanya perilaku (*behaviour*) **tambahan** (optional)
- Hubungan **include** mendefinisikan hubungan **langsung** dua *use-case* (*wajib dilibatkan*)



use-case B mungkin dilakukan bersamaan dengan *use-case* A



use-case B **dilibatkan** ketika *use-case* A dilakukan

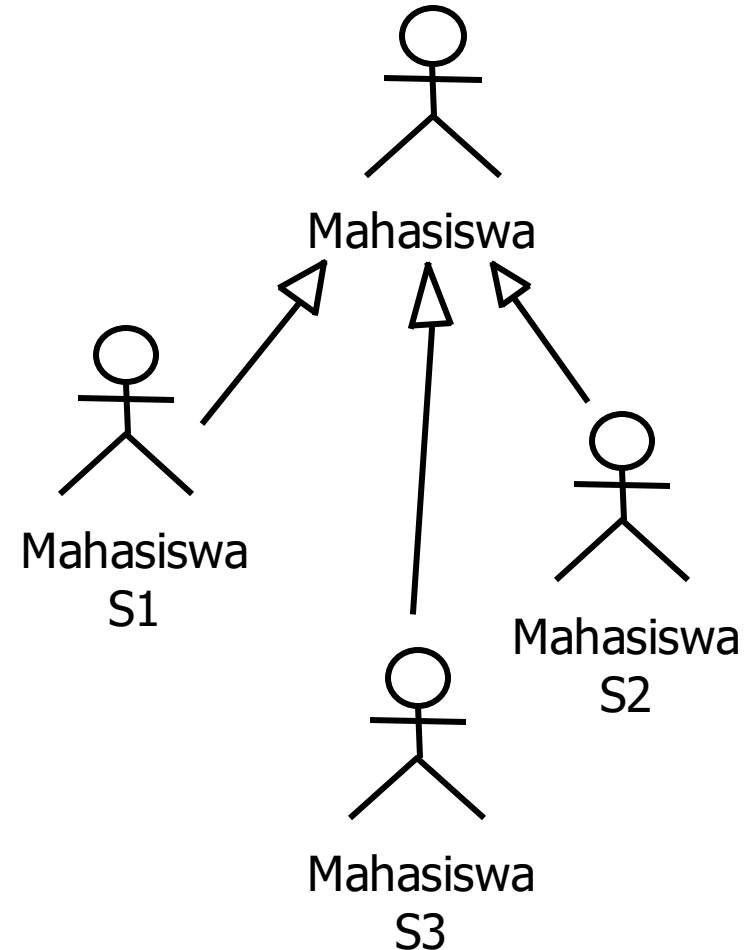
Extend/Include

- Dalam model *use-case*, dimungkin adanya pilihan *event* alternatif ataupun ada *use-case* yang harus dilakukan jika suatu *use-case* dieksekusi
 - **Alternatif** *use-case* ditunjukkan sebagai **<<extends>>**
 - Bagian yang **optional**
 - *Use-case* yang **harus** dilakukan ditunjukkan sebagai **<<include>>**
 - **Harus dilakukan**
- Walau jarang terjadi, tetapi *use-case extends* dan *include* bisa terjadi.
- Kenapa ada *extends/include*?
 - Karena harus ditambahkan fitur/ fungsionalitas baru
 - Untuk mendapatkan aksi tambahan dalam suatu kondisi lain

Generalisasi Aktor

(Actor Generalization/Actor Inheritance)

- Beberapa aktor dapat memiliki **peran** yang **sama** pada suatu *use-case*
- Contoh:
 - Ada mahasiswa S1, S2 dan S3 yang ketiganya terdaftar suatu kuliah
 - Ketiga akan terlihat sebagai entitas eksternal oleh *use-case* 'Daftar Ulang' atau 'Ambil mata kuliah'
 - Mahasiswa S1, S2 atau S3 dapat dimodelkan sebagai 'Mahasiswa' saja, karena ketiganya memiliki banyak kesamaan
- Hal ini disebut '**generalisasi aktor**'



Langkah pemodelan berbasis skenario

1. Identifikasi aktor
2. Identifikasi *use case*
3. Gambarkan diagram *use case*
4. Buat skenario tiap *use case*

1. Identifikasi aktor

- Siapa '**pengguna**' sistem atau yang terkait dengan sistem
 - Ada aktor yang **menggunakan** sistem
 - Ada aktor yang bertugas melakukan perawatan (***maintenance***)
- **Peran** aktor harus **berbeda**
 - Mungkin bisa terjadi peran yang saling tumpang tindih; modelkan dengan baik
 - Perlu **nama** yang 'relevan' dengan makna semantik dari peran
 - **Pengguna sistem** dengan **pelanggan sistem**, harus jelas apa yang dimaksud dengan peran ini.
 - Istilah **pengguna sistem** mungkin juga adalah **sistem lain** (bukan orang).
 - Juga harus jelas apa '**kebutuhan**' dan '**tanggung jawab**' si aktor!

2. Identifikasi use-case

- Daftarkan **aktivitas** yang dilakukan oleh **aktor** untuk melakukan suatu **fungsi/kegiatan**
- Beri **nama**
 - Nama menggambarkan sekumpulan aksi
 - Biasanya dimulai dengan '**kata kerja**'
- *Use-case* biasanya harus '**lengkap**' atau dapat '**berdiri sendiri**'
- *Use-case* ini memberikan suatu '**hasil**' untuk **aktor** ini

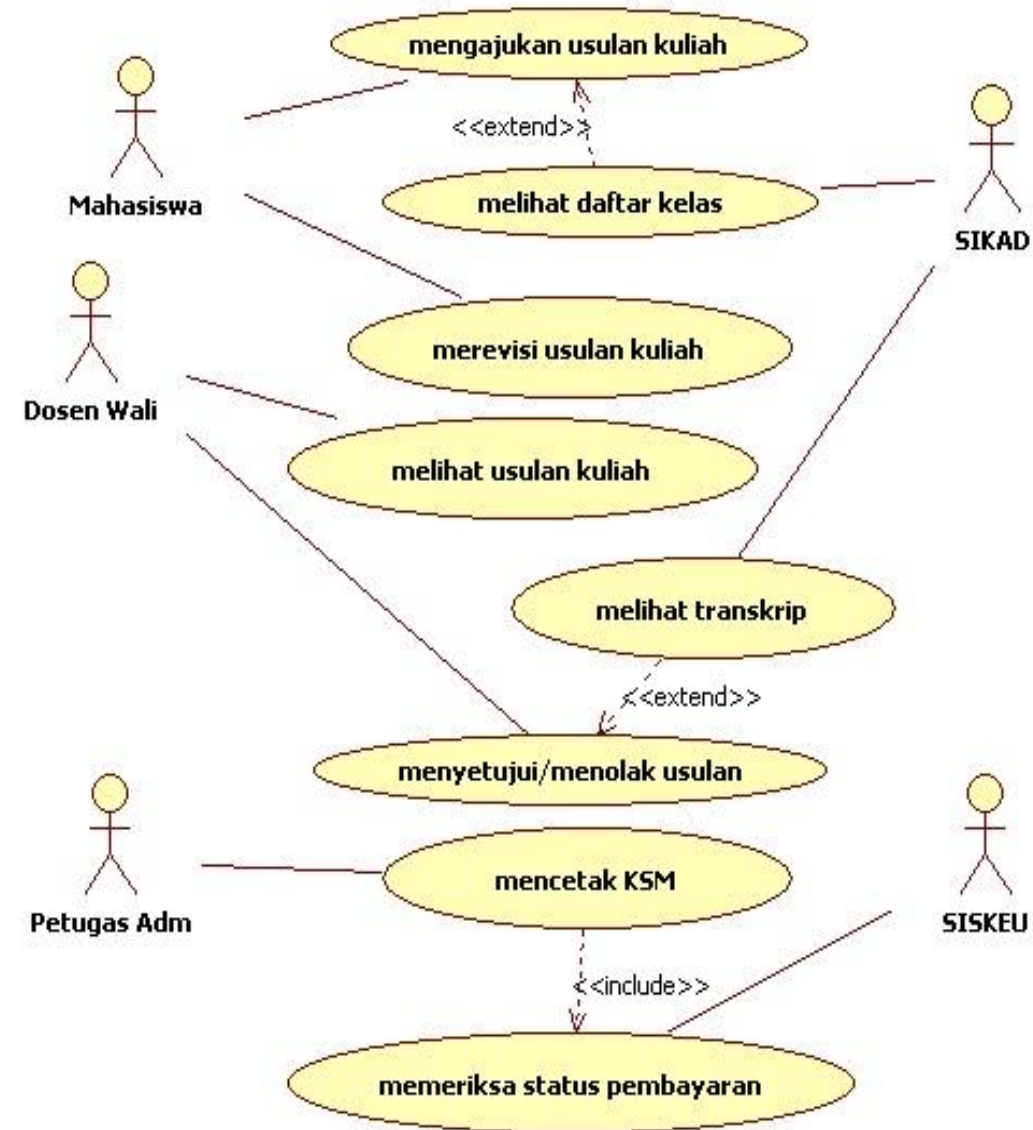
3. Gambarkan diagram use case

- Setiap ***use case*** harus **terhubung** dengan minimal **satu aktor**
- Jika *use case* terhubung dengan **lebih** dari satu aktor, maka harus diperjelas aktor yang men-***trigger*** *use case* pertama kali (gunakan **tanda panah**)
- Mungkin ada hubungan antar *use case*: ***include*** (*uses*) atau ***extends*** (*insert*)

4. Membuat skenario

- Skenario digunakan untuk **menjelaskan pemakaian sistem**
- Setiap skenario **dilihat dari sudut pandang ‘aktor’**
- Setiap skenario **menjawab pertanyaan** berikut:
 - Siapa aktor utama, siapa aktor pendukung
 - Apa tujuan si aktor
 - Kondisi awal apa yang harus ada sebelum suatu cerita atau ‘*story*’ dimulai
 - Apa tugas/fungsi utama yang dilakukan oleh si aktor
 - Apa tugas/fungsi tambahan yang dapat diberikan
 - Apa variasi yang memungkinkan dalam interaksi si aktor
 - Informasi dari sistem apa yang dibutuhkan, diproduksi atau diubah dari/oleh si aktor
 - Apakah si aktor harus memberitahukan sistem bila terjadi perubahan pada lingkungan eksternal?
 - Informasi apa yang diinginkan oleh si aktor dari sistem
 - Apakah si aktor ingin diberitahu bila ada perubahan yang di luar rencana?

Contoh Use-case Diagram



Skenario Use Case Mengajukan Usulan

Aksi Aktor (Mahasiswa)	Reaksi P/L
1. Memilih menu entri usulan kuliah	
	2. Menampilkan form entri FRS
3. Mengisikan kode kuliah	
	4. Menampilkan informasi detil matakuliah (nama, sks)
5. Menekan tombol SIMPAN	
	6. Menyimpan data usulan ke dalam basisdata

Alternatif skenario Mengajukan Usulan dan Melihat Daftar Kelas

Actor (Mahasiswa)	Reaksi P/L	Actor (SIKAD)
1. Memilih menu entri usulan kuliah		
2. Memilih untuk melihat daftar kelas		
	3. Meminta daftar kelas ke SIKAD	
		4. Mengirim daftar kelas
	5. Menampilkan daftar kelas yang dibuka	
6. Memilih matakuliah dari daftar		
7. Menekan tombol SIMPAN		
	8. Menyimpan data usulan ke dalam basisdata	

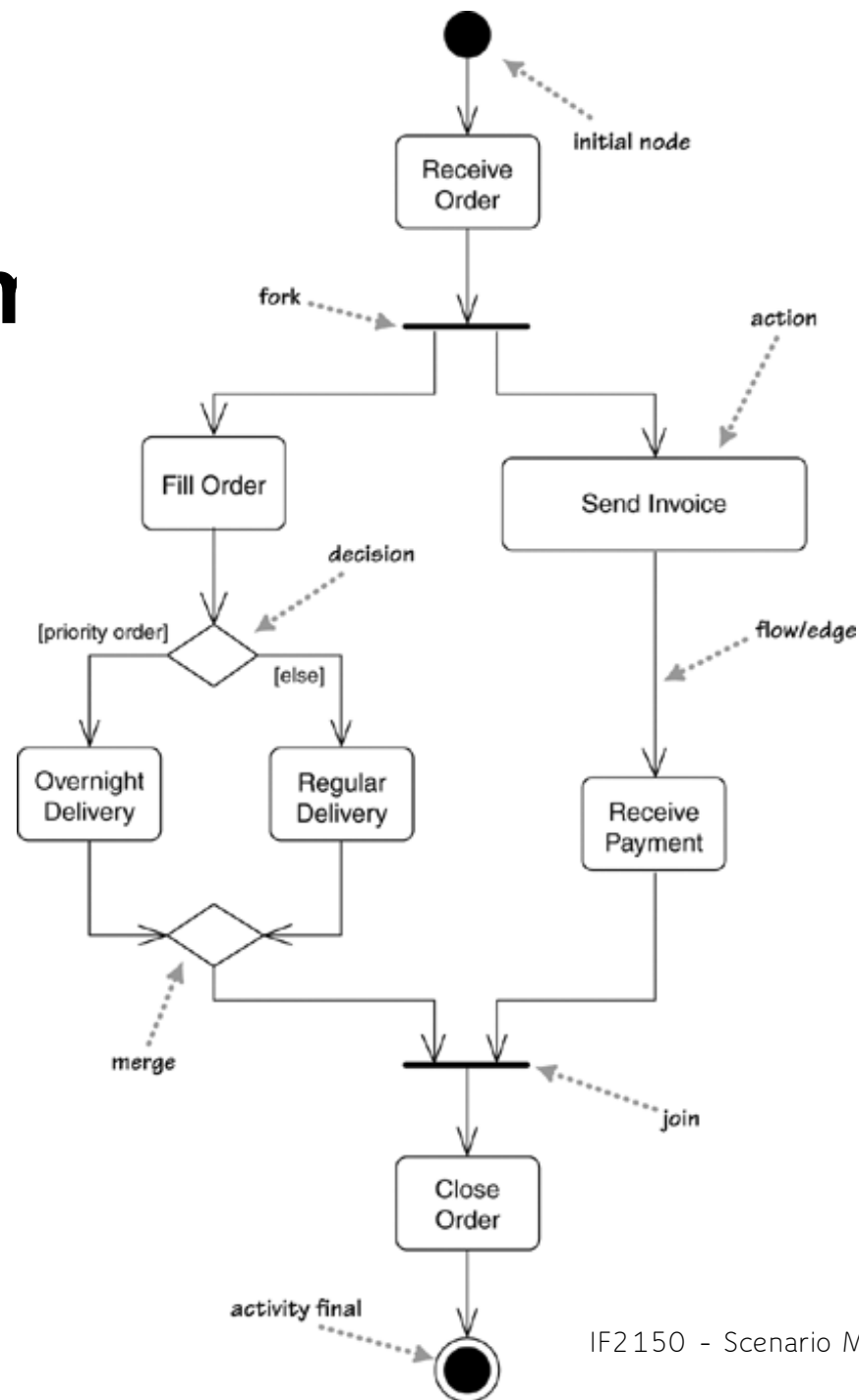
Deskripsi Use Case

- Untuk jumlah *use-case* yang **besar** dengan berbagai **alternatif**, maka penulisan teks menjadi tidak praktis, jadi kadang digunakan **diagram**
 - **Diagram Statechart**
 - Berisi penjelasan *state* dan transisi dalam *use-case*
 - **Diagram Aktivitas**
 - Menggambarkan urutan aksi
 - **Diagram Interaksi**
 - Menjelaskan interaksi antar instansiasi dari aktor dan instansiasi dari *use-case*

Activity Diagram

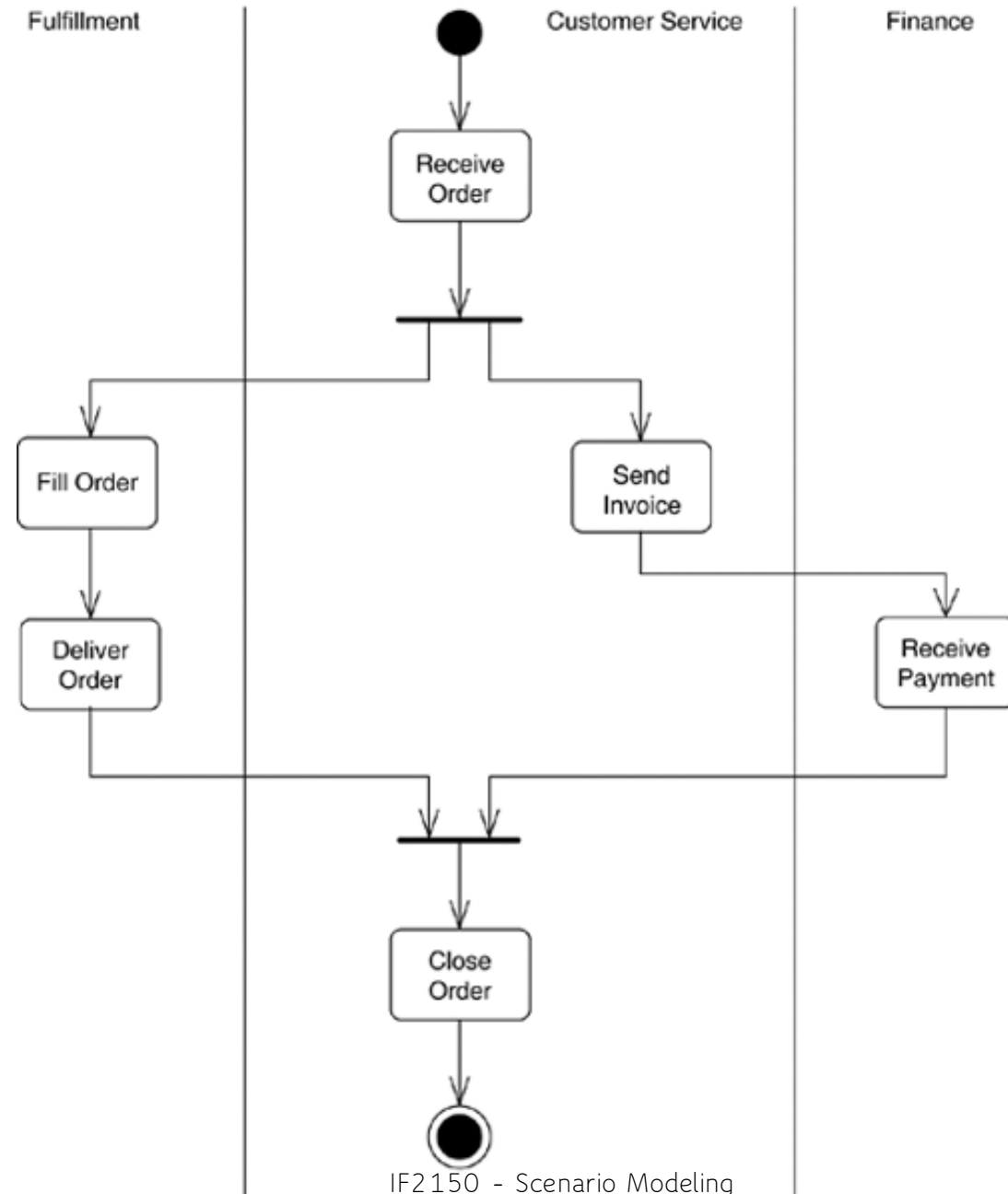
- Diagram aktivitas menjelaskan
 - Urutan proses prosedural
 - Urutan bisnis proses
 - Urutan kerja (*work flow*)
- Diagram aktivitas ini mirip seperti '*Flow chart*'
 - Tetapi *flow-chart* tidak mengenal 'perilaku paralel'/'konkuren'

Contoh Activity Diagram



Partisi

Diagram aktivitas dapat dipartisi berdasarkan 'siapa' yang melakukan 'apa'



Perhatian:

- Use case diagram tidak menggambarkan urutan proses
- Use case adalah **kumpulan aksi**
 - perhatikan granularitas use case → jangan terlalu kecil
 - realisasi use case akan melibatkan **beberapa kelas**
- Jelaskan singkat setiap **use case** dan **actor**
- Jelaskan singkat diagram use case
- Lengkapi skenario untuk setiap use case
 - Skenario normal
 - Skenario alternatif