



Rekayasa Perangkat Lunak

USE CASE dan DESKRIPSI USE CASE



PROSES PERANCANGAN PL

Fase	Kegiatan	Hasil
Analysis	Menentukan kebutuhan sistem	software requirements specification
Design	Mengembangkan struktur sistem	Software design description
Implementation	Menulis source code	program, file, modul
Testing	Menguji sistem yang dibuat/dikembangkan	test suits dan hasil test
Maintenance	koreksi, adaptasi, improve	Versi baru sistem



Tujuan Analisis Sistem

- Menghasilkan perangkat lunak yang bebas kesalahan (*fault-free software*)
- Selesai tepat waktu, sesuai dengan anggaran
- Memenuhi kebutuhan pemakai



UML

- UML adalah permodelan visual untuk melakukan desain perangkat lunak.
- UML dapat di implementasikan di beberapa area yang berbeda dan dapat mengkomunikasikan semua hal, mulai dari proses bisnis sampai dengan pendistribusian perangkat lunak
- UML Provides a language to capture information that varies greatly depending on the domain of the problem. In doing that, there are often parts of UML that either don't apply to your particular problem or may not lend anything to the particular view you are trying to convey. It is important to realize that you don't need to use every part of UML in every model you create.

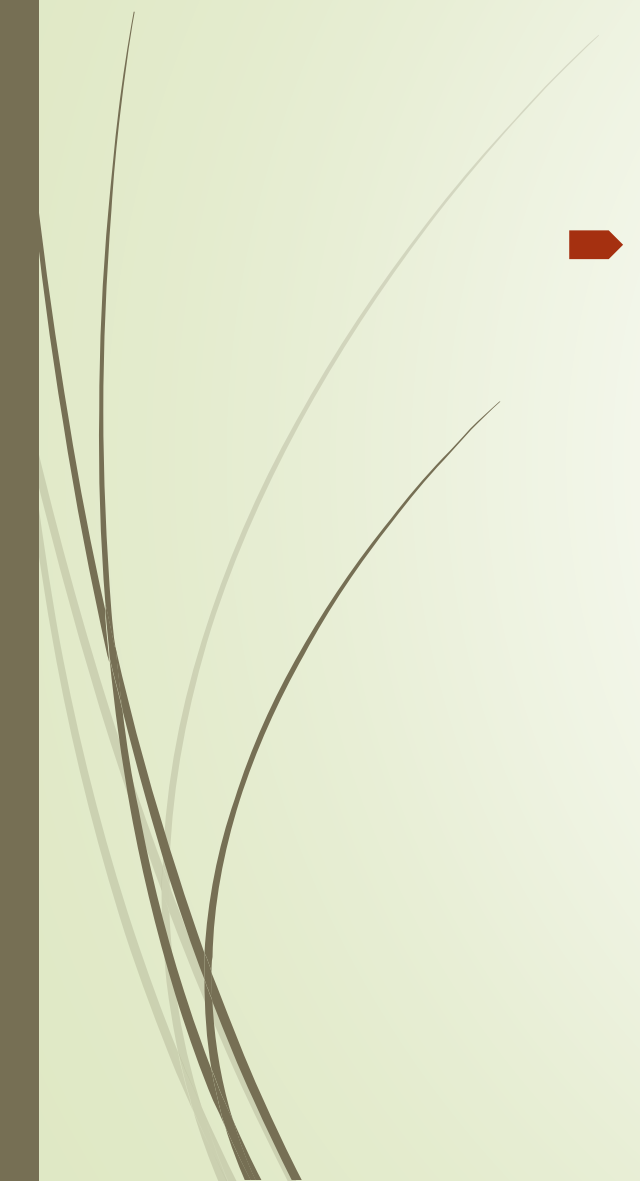


USE CASE DIAGRAM

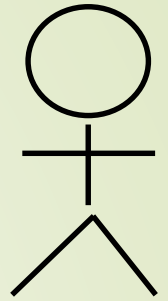
- Gambaran fungsionalitas dari sistem
- Menekankan “apa” bukan “bagaimana”
- Melihat kebutuhan system dari sudut pandang user
- Menggambarkan hubungan antara use case dengan actor
- Use case :
 - Pola perilaku system
 - Urutan transaksi yang berhubungan yang dilakukan oleh satu aktor



USE CASE DIAGRAM

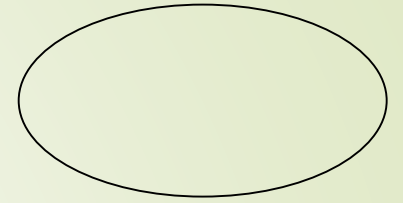
- Use case terdiri dari :
 - Use case
 - Actors
 - Relationship
 - System boundary boxes (optional)
 - Packages (optional)
- 

ACTOR



- Actor adalah orang, system, atau external entitas yang menyediakan atau menerima informasi
- Actor menggambarkan sebuah tugas / peran bukan posisi atau jabatan
- Actor memberi input atau menerima informasi dari system
- Actor biasanya menggunakan kata benda
- Tidak boleh ada komunikasi langsung antar actor

USE CASE

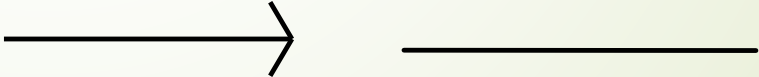


- Berdasarkan keperluan Actor. Apa yang dikerjakan sistem, bukan bagaimana sistem mengerjakan
- Use Case biasanya menggunakan kata kerja
- Nama Use Case harus unik dan tidak boleh ada 2 Use Case yang memiliki nama yang sama.

ASSOCIATION

- Association bukan menggambarkan aliran data / informasi
- Association digunakan untuk menggambarkan bagaimana actor terlibat dalam use case

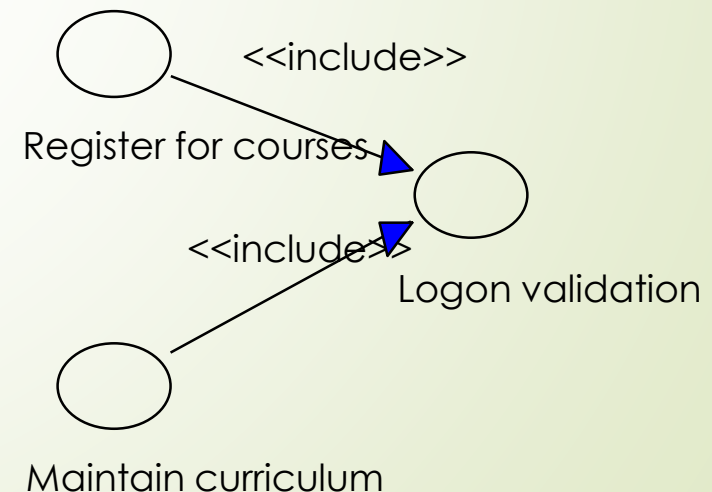
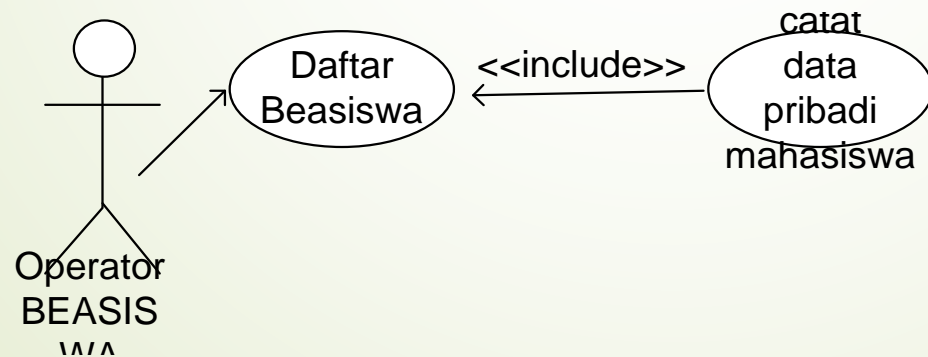
ASSOCIATION ACTOR – USE CASE

- Ujung panah pada association antara actor dan use case mengindikasikan **siapa/apa** yang meminta interaksi dan bukannya mengindikasikan aliran data
- Association antara actor – use case menggunakan panah terbuka atau garis tanpa panah.
- Dilambangkan dengan 

ASSOCIATION ANTAR-USE CASE

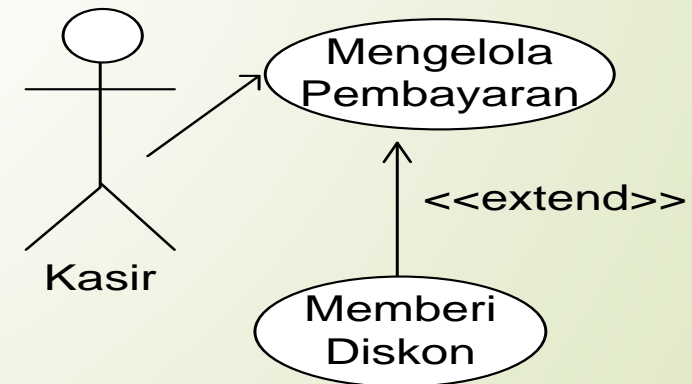
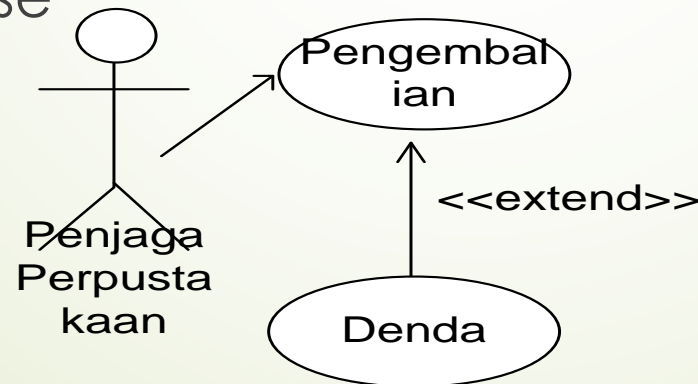
- `<<include>>` termasuk didalam use case lain (required) / (diharuskan)
- Pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program
- Tanda panah terbuka harus terarah ke sub use case

Contohnya :




ASSOCIATION ANTAR-USE CASE (2)

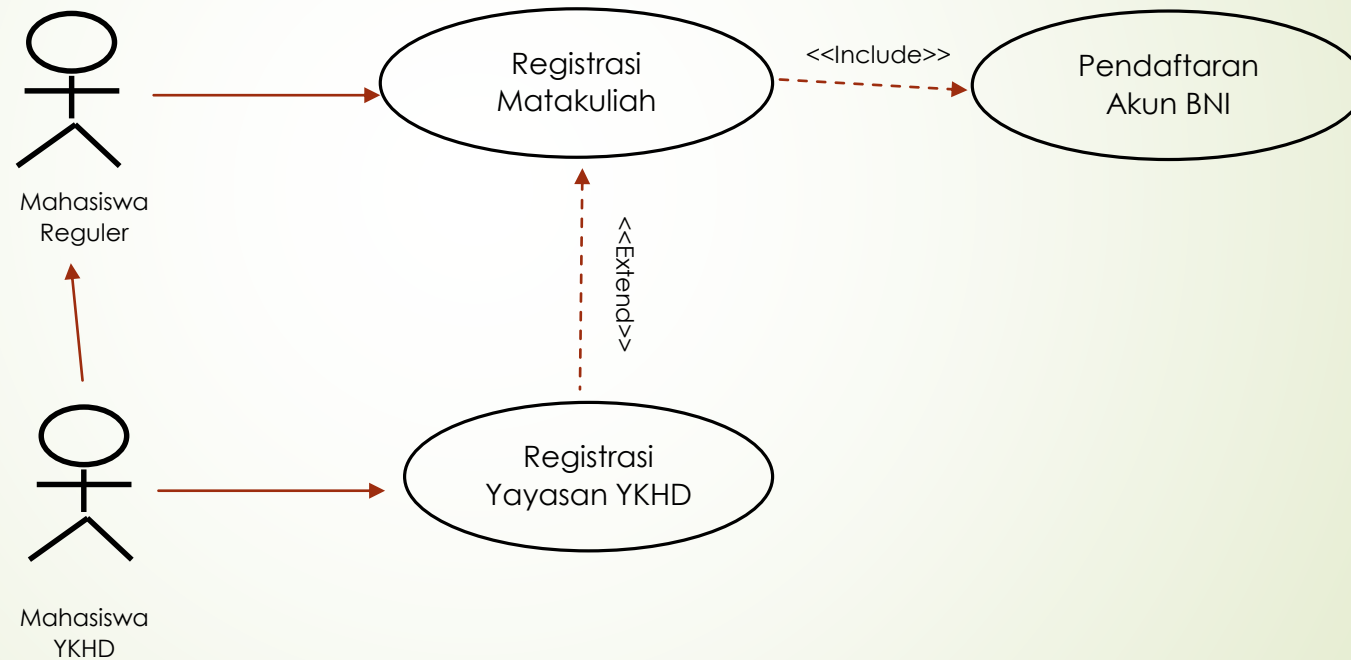
- <<extend>> perluasan dari use case lain jika kondisi atau syarat terpenuhi
- Kurangi penggunaan association Extend ini, terlalu banyak pemakaian association ini membuat diagram sulit dipahami.
- Tanda panah terbuka harus terarah ke parent/base use case



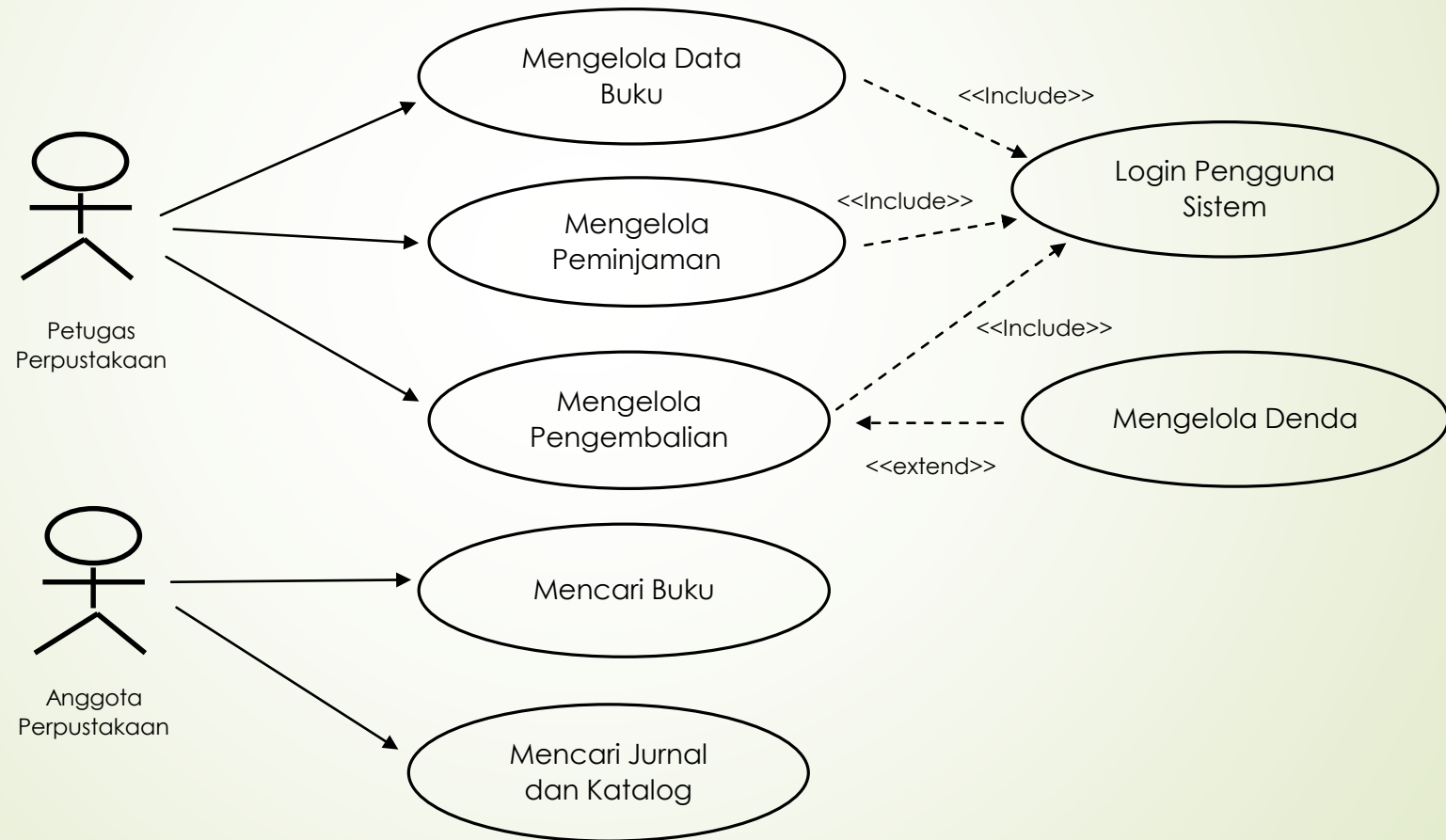
INHERITANCE ANTAR-USE CASE

- Generalization/inheritance digambarkan dengan sebuah garis berpanah tertutup pada salah satu ujungnya yang menunjukkan lebih umum
- Gambarkan generalization/inheritance antara use case secara vertical dengan inheriting use case dibawah base/parent use case
- Generalization/inheritance dipakai ketika ada sebuah keadaan yang lain sendiri/perlakuan khusus (*single condition*)
- Dilambangkan dengan 

INHERITANCE ANTAR-ACTOR



CONTOH USE CASE





Deskripsi Use Case

- Deskripsi Use Case digunakan untuk melakukan deskripsi terhadap interaksi yang terjadi antara sistem dan aktor yang terjadi dalam Use Case
- Terdapat Template untuk Deskripsi Use Case
 - Nama Use Case
 - Actor
 - Deskripsi Singkat
 - Pre Condition
 - Flow Of Event
 - Post Condition



Pre Condition Use Case

- Pre condition (pra kondisi) menyatakan (pra syarat) apa yang harus ada sebelum use case dijalankan.
- Pre condition harus benar atau terpenuhi supaya fungsionalitas yang dinyatakan dalam use case bisa terpenuhi.

POST Condition

- Post condition menyatakan apa yang didapat atau terjadi setelah use case dijalankan.
- Post condition merupakan kondisi yang akan benar atau terpenuhi setelah fungsionalitas dijalankan.



Flow Of Events

- Menjelaskan langkah – langkah yang ada dalam tiap use case
- Di dalam memberikan penjelasan harus deklaratif dan diurutkan berdasarkan waktu kejadian
- Penjelasan harus dilihat berdasarkan sudut pandang actor
- Penjelasan di mulai oleh aktor yang memicu berjalannya use case
- Contoh :
 - Petugas Perpustakaan Mencari Data Anggota Perpustakaan
 - Petugas Perpustakaan Mencari Data Buku
- Penjelasan harus jelas dan tidak ambigu

Contoh Deskripsi Use Case

Nama Use Case		Mengembalikan Buku
Aktor		Petugas Perpustakaan
Deskripsi Singkat		Use Case Menjelaskan Bagaimana Alur Pengembalian Buku
Pre Condition		Data Peminjaman sudah tercatat ke dalam sistem
Flow of Event	1	Use case dimulai ketika peminjam datang dengan membawa buku yang akan dikembalikan.
	2	Petugas memasukkan ID peminjam
	3	Sistem menampilkan data peminjam beserta dengan daftar buku yang dipinjam
	4	Petugas mencari buku yang akan dikembalikan dari daftar buku yang ada
		<overduebook>
	4	Petugas menghapus buku dari daftar peminjaman
	5	Use case selesai
Post Condition		Buku sudah dikembalikan

Contoh Deskripsi Use Case

Nama Use Case		Mengeluarkan Denda
Aktor		Petugas Perpustakaan
Deskripsi Singkat		Use case mendeskripsikan bagaimana Petugas Mengeluarkan Denda untuk buku yang telat dikembalikan
Pre Condition		---
Flow of Event	1	Use case dimulai ketika petugas mendapatkan bahwa buku yang dikembalikan melebihi batas waktu peminjaman
	2	Petugas meminta sistem untuk menghitung denda yang dikenakan.
	3	Sistem menghitung denda
	4	Sistem mencetak denda yang harus dibayar oleh peminjam
	5	Use case selesai
Post Condition	



Discussion Time

- Buatlah Use Case dan Deskripsi Use Case dari Aplikasi yang akan dibuat