

# PERTEMUAN 5

## Use Case Diagram

# Use Case Diagrams

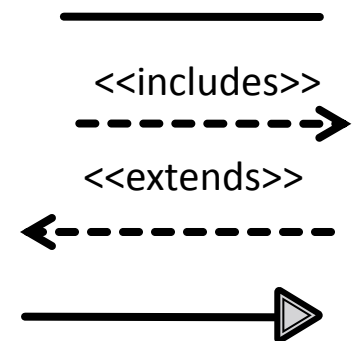
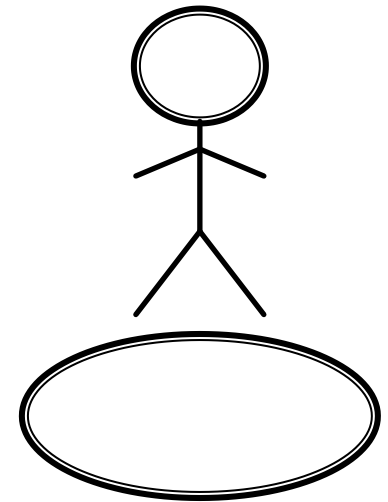
- Summarized into a **single picture**
- All of the use cases for the part of the system being modeled
- Use case represents the discrete **activities performed by the user**
- Use Case Diagram tells **what the system will do**
- Good for **communicating with users**

# Deskripsi USE CASE

- Sebuah use case adalah situasi dimana sistem digunakan untuk memenuhi satu atau lebih kebutuhan pemakai.
- Use case merupakan awal yang sangat baik untuk setiap fase pengembangan berbasis objek, design testing, dan dokumentasi.
- Use case menggambarkan kebutuhan sistem dari sudut pandang di luar sistem.
- Use case menentukan nilai yang diberikan sistem kepada pemakainya.
- Use cases hanya menetapkan apa yang seharusnya dikerjakan oleh sistem, yaitu kebutuhan fungsional sistem.
- Use case tidak untuk menentukan kebutuhan nonfungsional, misalnya: sasaran kinerja, bahasa pemrograman, dsb.

# Syntax for an Use Case Diagram

- **Actor**
  - person or system that derives benefit from and is external to the subject
- **Use Case**
  - Represents a major piece of system functionality
- **Association Relationship**
- **Include Relationship**
- **Extend Relationship**
- **Generalization Relationship**



# Use Case

- A major piece of **system functionality**
- Can **extend** other Use Cases
- Placed inside system boundary
- Labeled with descriptive **verb - noun phrase**



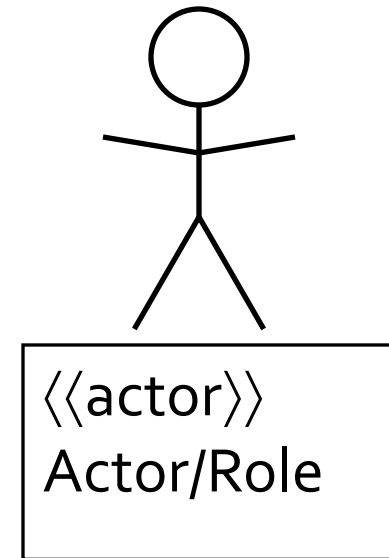
# System Boundary

- Includes the **name of the system** inside or on top
- Represents the **scope of the system**
- **Actors are outside** the scope of the system

Boundary

# Actor

- A **person** or **another system** that interacts with the current system
- A **role**, not a specific user
- **Provides input, receives output**, or both



# Association Relationship

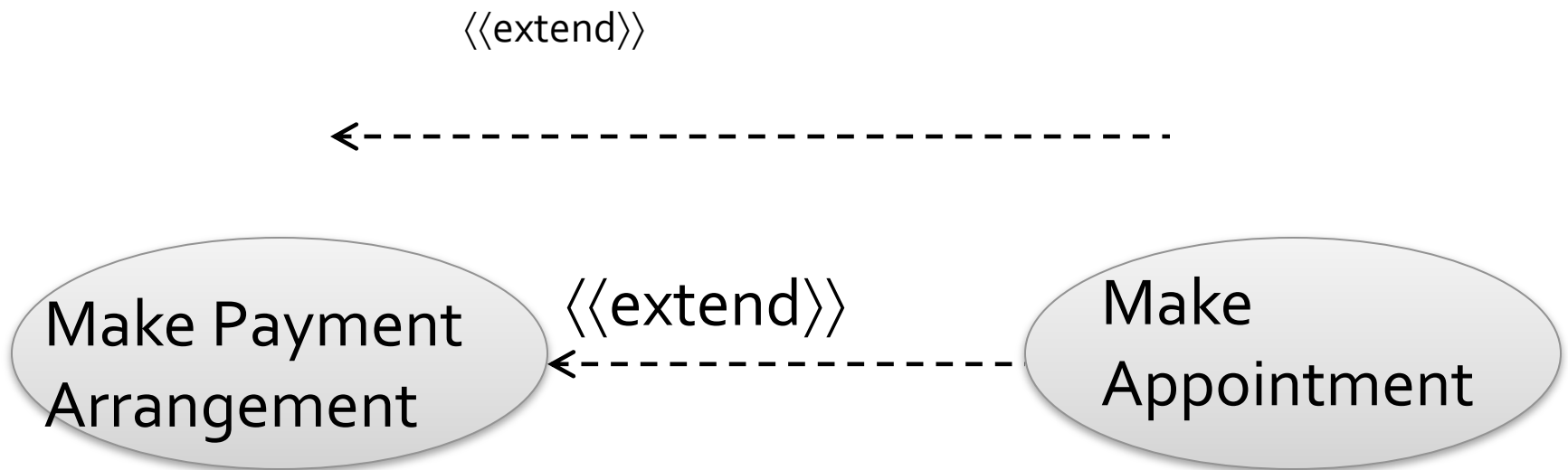
- **Links** actor and the Use Case
- Shows **two-way communication**
  - If one-way, arrows are used
- \* is for "**multiplicity** of the Association"





# Extends Relationship

- **Extends** Use Case to include **Optional** behavior
- **Arrow points** from the extension Use Case **to** the base Use Case



# Include Relationship

- Include one Use Case from within another
- Arrow points from base Use Case to the included Use Case

⟨⟨include⟩⟩

←-----

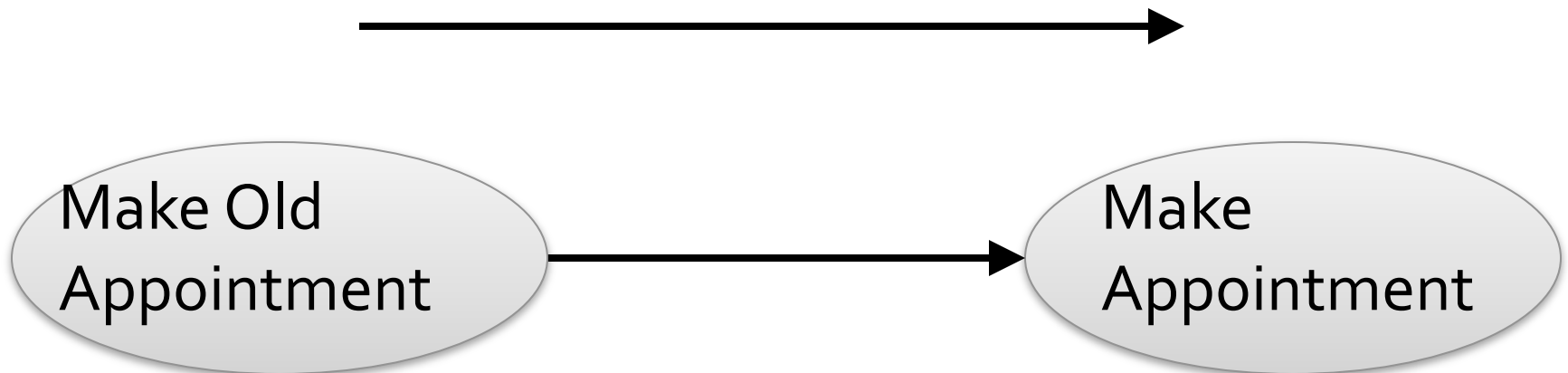
Make New  
Patient Appointment

⟨⟨include⟩⟩

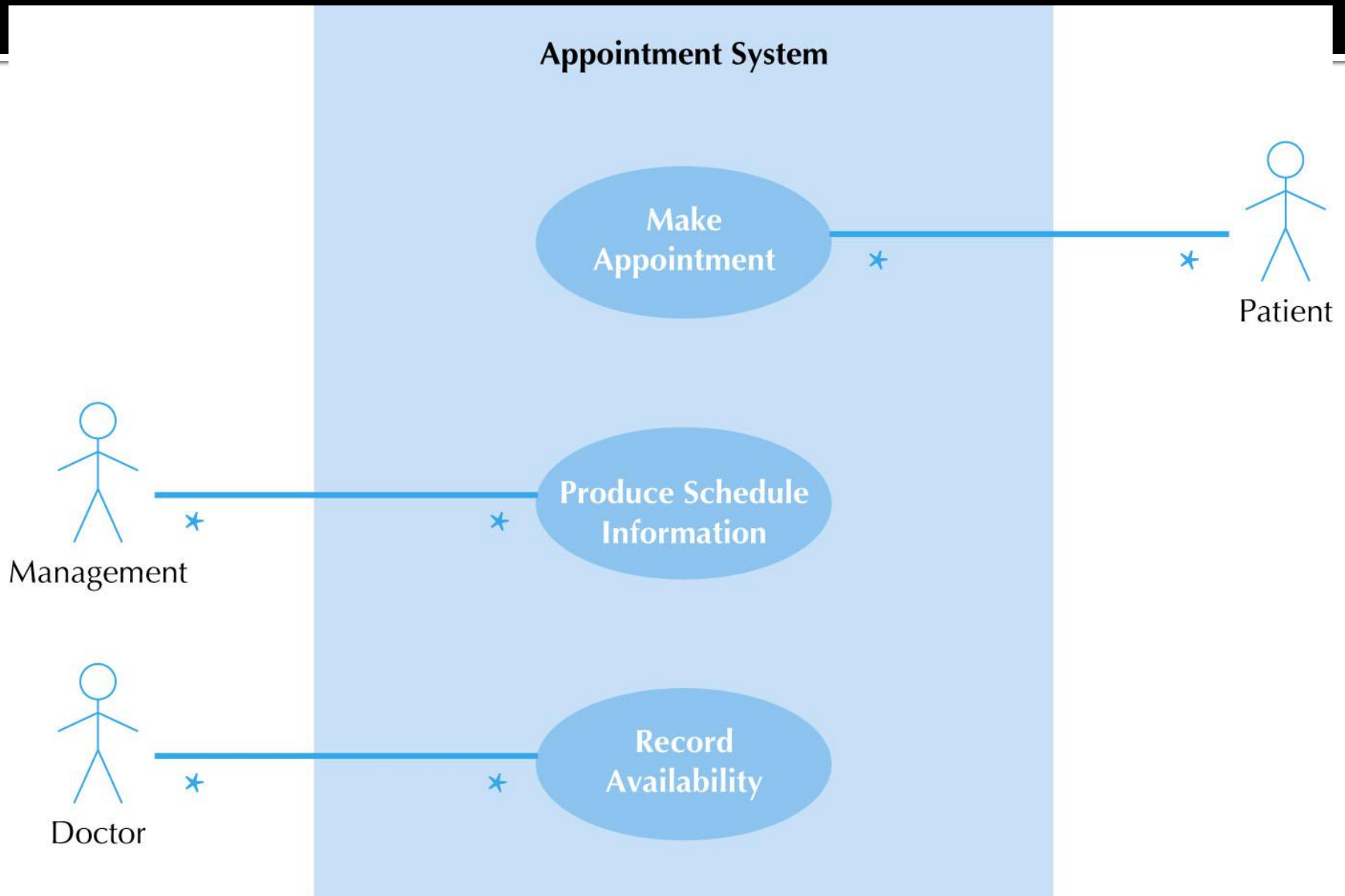
Create New  
Patient

# Generalization Relationship

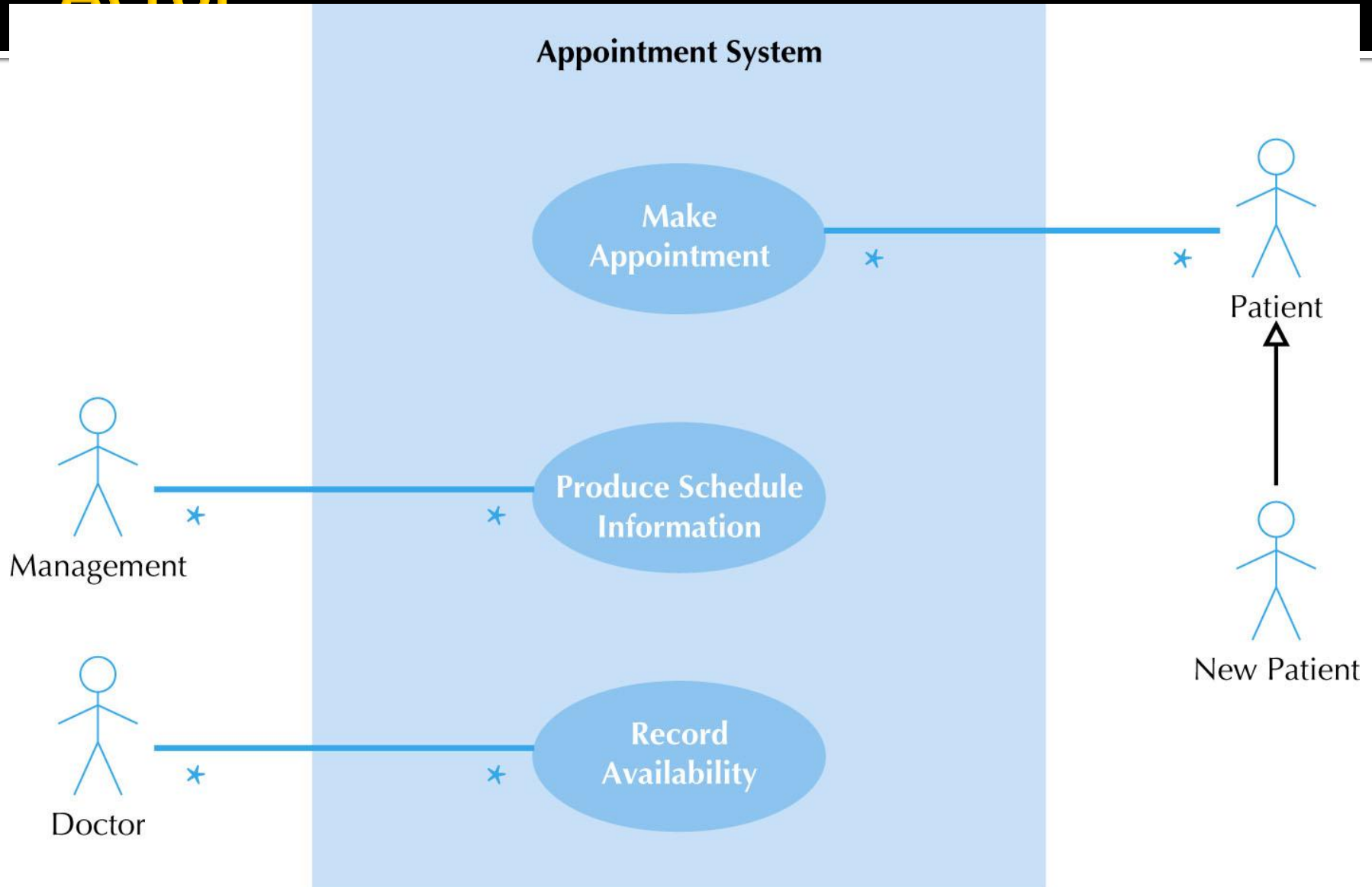
- A specialized Use Case to a more generalized Use Case
- **Arrow points** from specialized **to general** Use Case



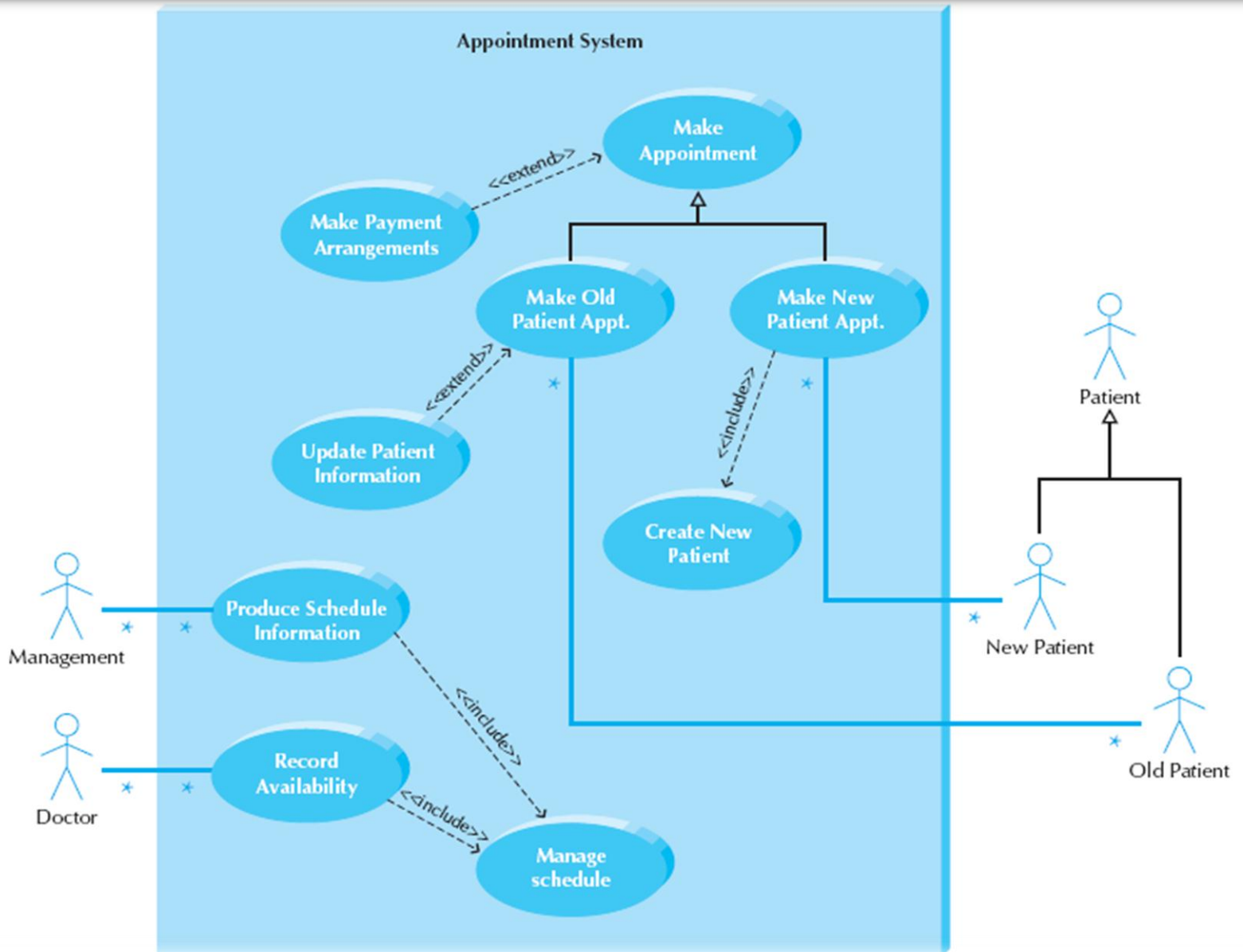
# Use Case Diagram for Appointment System



# Use Case Diagram with Specialized Actor



# Extend and Include Relationships



# USE CASE Diagram

Use case diagram terdiri dari

**a. Use case**

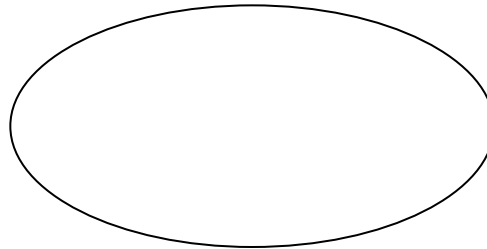
- Use case class digunakan untuk memodelkan dan menyatakan unit fungsi / layanan yang disediakan oleh sistem (or bagian sistem: subsistem atau class) ke pemakai.
- Use case dapat dilingkupi dengan batasan sistem yang diberi label nama sistem.
- Use case adalah sesuatu yang menyediakan hasil yang dapat diukur ke pemakai atau sistem eksternal.

- Use case dibuat berdasar keperluan actor, merupakan “apa” yang dikerjakan system, bukan “bagaimana” system mengerjakannya
- Use case diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya dengan actor.
- *Use case* dinotasikan dengan gambar (horizontal ellipse).
- Use case biasanya menggunakan kata kerja.
- Nama use case boleh terdiri dari beberapa kata dan tidak boleh ada 2 use case yang memiliki nama yang sama



- Use case class memiliki objek use case yang disebut skenario. Skenario menyatakan urutan pesan dan tindakan tunggal.

- Simbol use case :



## **b. Actors**

- *Actor* menggambarkan orang, system atau external entitas / stakeholder yang menyediakan atau menerima informasi dari system
- *Actor* menggambarkan sebuah tugas/peran dan bukannya posisi sebuah jabatan
- *Actor* memberi input atau menerima informasi dari system
- *Actor* biasanya menggunakan Kata benda
- Actor adalah eksternal terhadap sistem.

- Tidak boleh ada komunikasi langsung antar actor
- Indikasi <<system>> untuk sebuah actor yang merupakan sebuah system
- Adanya actor bernama “Time” yang mengindikasikan scheduled events (suatu kejadian yang terjadi secara periodik/bulanan)
- Letakkan actor utama anda pada pojok kiri atas dari diagram

- Actor berinteraksi dengan sistem.
- Actor memanfaatkan fungsi yang disediakan sistem, termasuk fungsi aplikasi dan pemeliharaan.
- Actors bisa saja menyediakan fungsi ke sistem.
- Actors bisa menerima informasi yang disediakan sistem.
- Actors bisa menyediakan informasi ke sistem.

➤ Simbol actor :

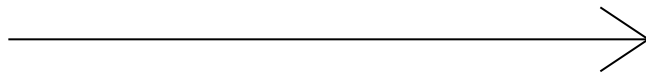


## c. Association

- Associations bukan menggambarkan aliran data/informasi
- Associations digunakan untuk menggambarkan bagaimana actor terlibat dalam use case
- Ada 4 jenis relasi yang bisa timbul pada use case diagram
  1. Association antara actor dan use case
  2. Association antara use case
  3. Generalization/Inheritance antara use case
  4. Generalization/Inheritance antara actors

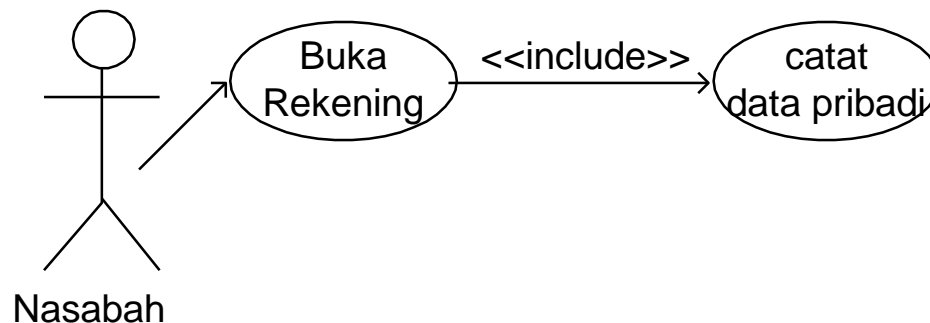
## 1. Association antara actor dan use case

- Ujung panah pada association antara actor dan use case mengindikasikan ***siapa/apa*** yang meminta interaksi dan bukannya mengindikasikan aliran data
- Sebaiknya gunakan **Garis tanpa panah** untuk association antara actor dan use case
- association antara actor dan use case yang menggunakan **panah terbuka** untuk mengindikasikan bila actor berinteraksi secara ***pasif*** dengan system anda



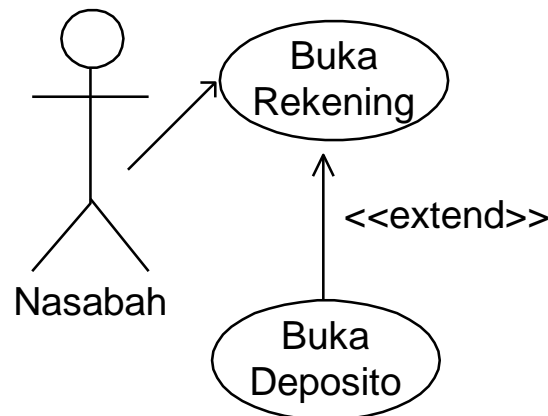
## 2. Association antara use case

- <<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
  - Gambarkan association include secara horizontal



## Association antara use case (Lanjut)

- <<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
  - Gambarkan association extend secara vertical

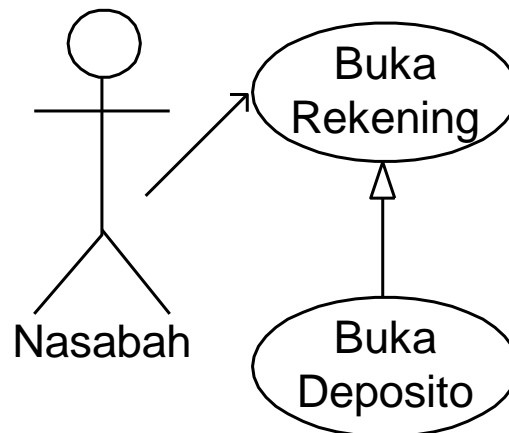




### 3. Generalization/inheritance antara use case

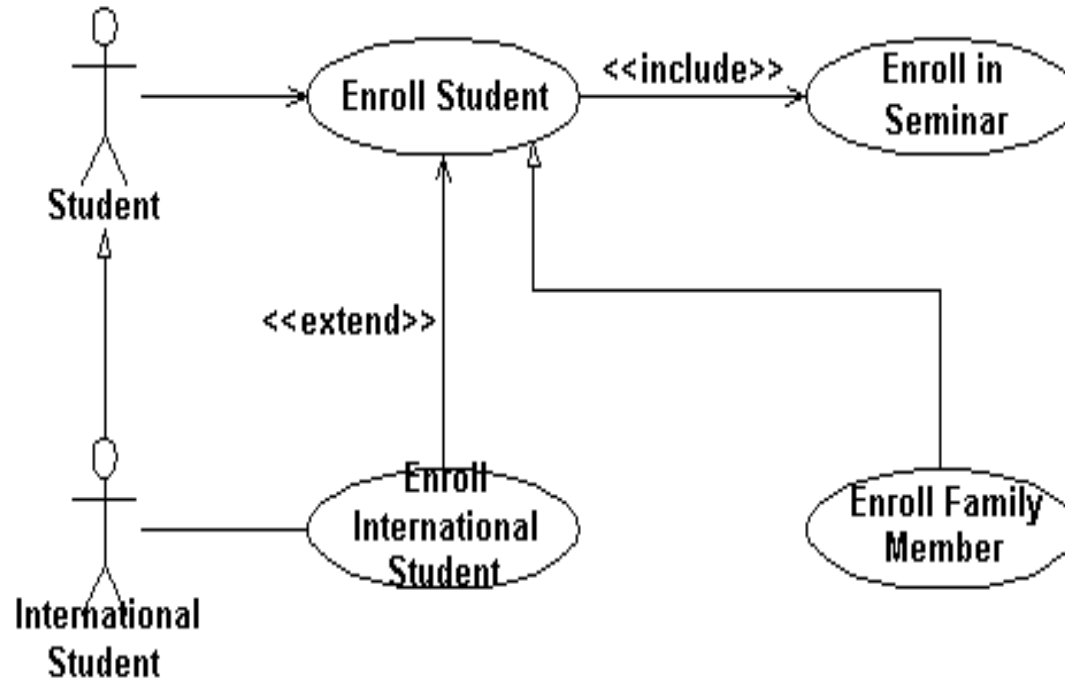
- Generalization/inheritance digambarkan dengan sebuah garis berpanah tertutup pada salah satu ujungnya yang menunjukkan lebih umum

- 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*)



#### 4. Generalization/inheritance antara actor

- Gambarkan generalization/inheritance antara actors secara vertical dengan inheriting actor dibawah base/parent use case



#### d. System boundary boxes (optional)

- Untuk memperlihatkan batasan sistem dalam diagram use case, Anda dapat menggambarkan sebuah kotak yang melingkupi semua use case, namun actor tetap berada di luar kotak
- Biasanya digunakan apabila memberikan beberapa alternative system yang dapat dijadikan pilihan.
- System boundary boxes dalam penggunaannya optional



# Studi Kasus: ATM System

## USE CASE DIAGRAM

# User Interface Design (Netbeans)

Menu Login

Masukkan PIN

Login

Menu Transaksi

Melihat Saldo

Mengirim Uang

Mengambil Uang

Keluar

# User Interface Design

Menu Utama

7	8	9
4	5	6
1	2	3
0	Enter	

Kotak Kartu

Kotak Uang

# ATM System



# ATM System

Layar

Kotak Uang

Kotak Kartu

Kotak Kuitansi



Masukkan PIN:

Kotak Uang

Kotak Kartu

Kotak Kuitansi

## Menu Utama

1. Melihat Saldo
2. Mentransfer Uang
3. Mengambil Uang
4. Logout

Kotak Uang

Kotak Kartu

Kotak Kuitansi

## Menu Melihat Saldo

1. Saldo anda adalah ....

Kotak Uang

Kotak Kartu

Kotak Kuitansi

## Menu Mentransfer Uang

### 1. No Account Penerima:

Kotak Uang

Kotak Kartu

Kotak Kuitansi

## Menu Mentransfer Uang

1. Jumlah uang yang dikirim:

Kotak Uang

Kotak Kartu

Kotak Kuitansi

## Menu Mentransfer Uang

### 1. Uang berhasil terkirim

Kotak Uang

Kotak Kartu

Kotak Kuitansi

## Menu Mengambil Uang

1. Jumlah uang yang diambil:

Kotak Uang

Kotak Kartu

Kotak Kuitansi

## Menu Mengambil Uang

Uang berhasil diambil

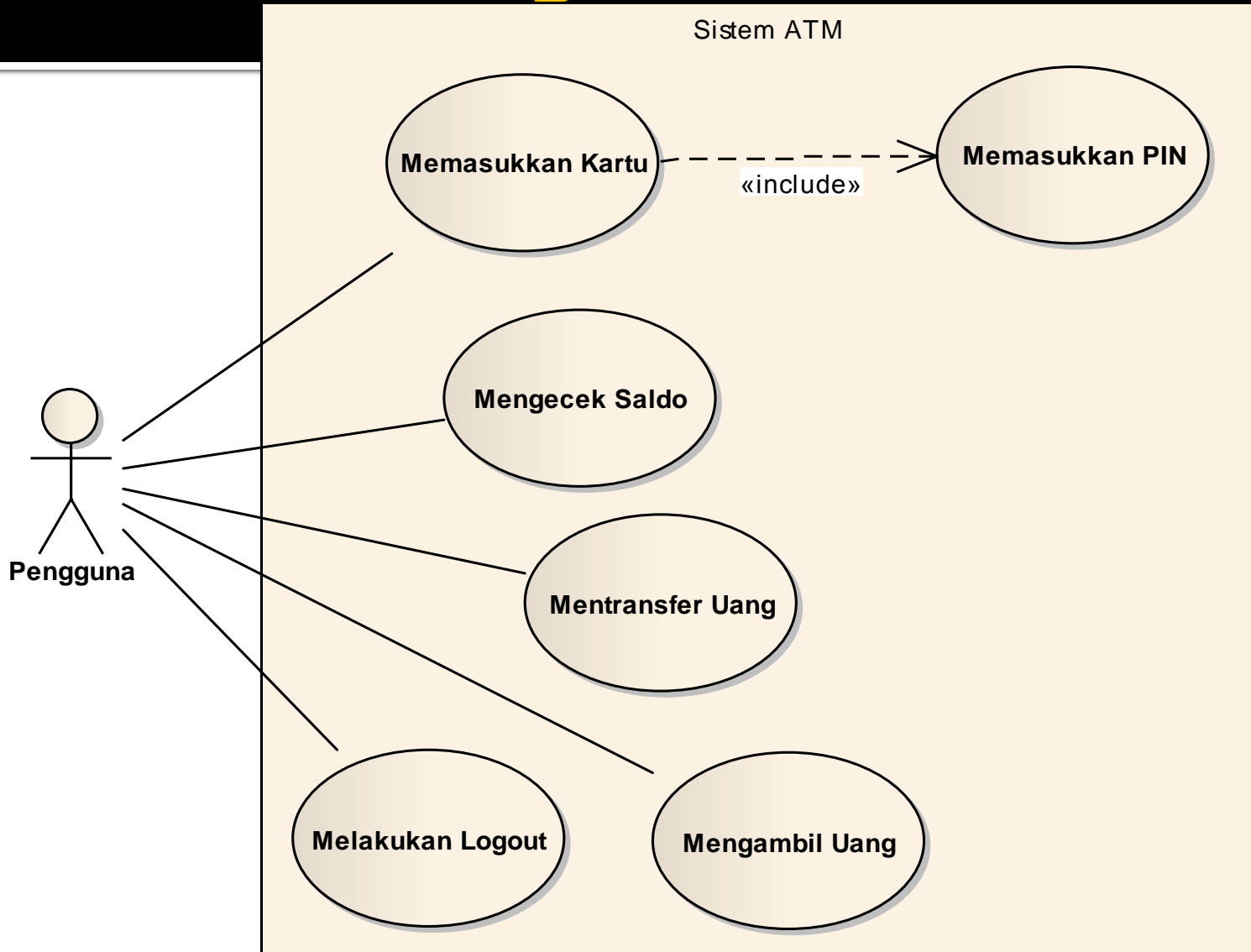
Kotak Uang

Kotak Kartu

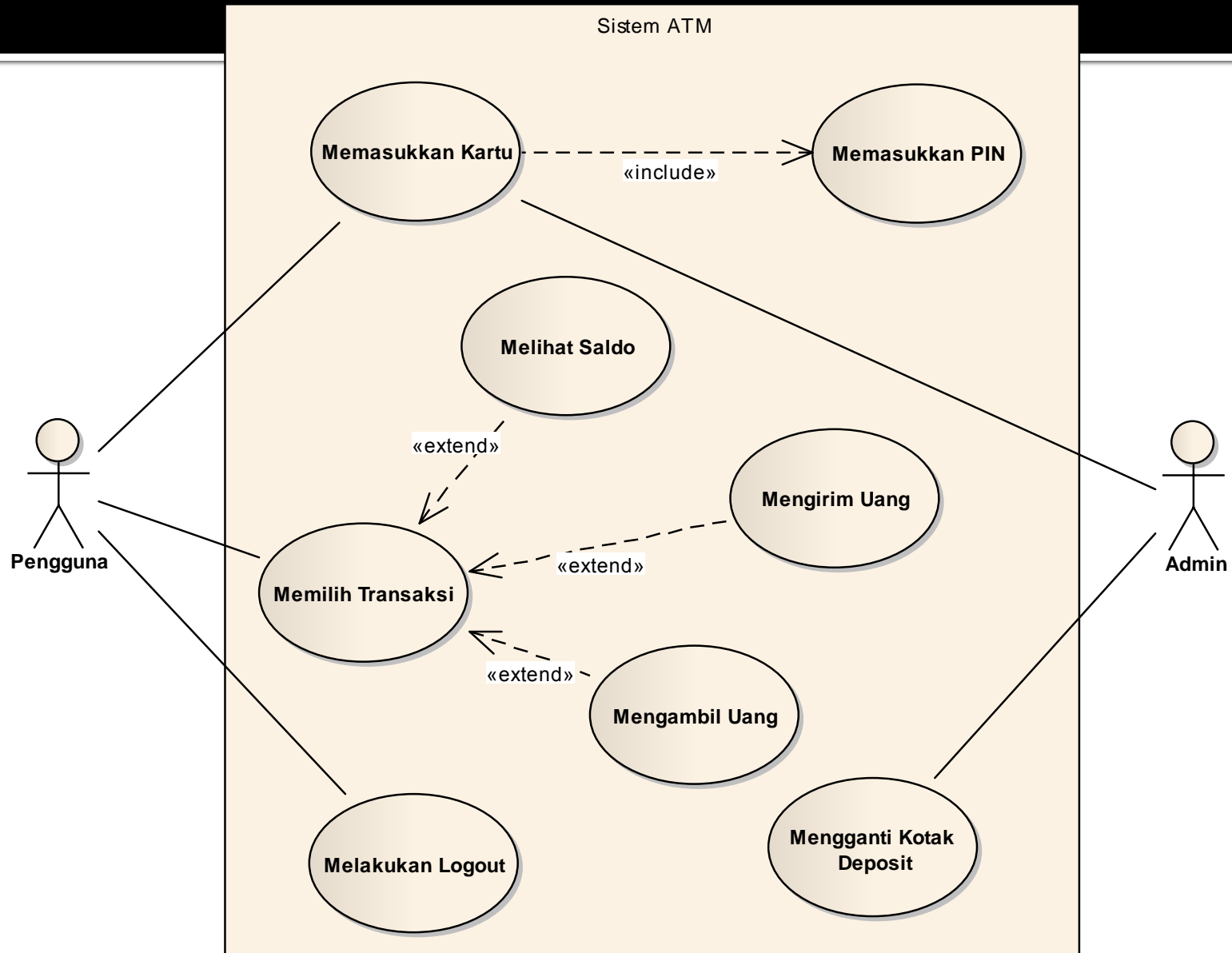
Kotak Kuitansi



# Use Case Diagram



# Use Case Diagram (Alternatif)



# LATIHAN : Buat Use Case Diagram Dan Activity Diagram Dari Skenario Di Bawah Ini

- Actor yang terlibat di dalam Sistem Informasi Perpustakaan ada pegawai/petugas disebut pustakawan dan anggotanya adalah mahasiswa. Pustakawan adalah operator sisfo yang diharuskan untuk login terlebih dahulu.
- Mahasiswa yang ingin meminjam buku diharuskan untuk melakukan registrasi terlebih dahulu dengan mengisi formulir pendaftaran. Kemudian data anggota akan diinputkan ke dalam sistem informasi dan akan diberikan kartu anggota oleh pustakawan.
- Mahasiswa yang ingin meminjam buku sebelumnya dapat melakukan pencarian buku dengan memasukkan data judul, pengarang, dan penerbit. Peminjaman mhs dapat meminjam maksimal 3 buku, untuk masa peminjaman selama 1 minggu dan transaksi peminjaman akan dimasukkan ke sistem informasi oleh pustakawan dengan membuka form pinjam, masukan id peminjam, tgl pinjam, dan data buku.



terlebih dahulu dengan mengisi formulir pendaftaran. Kemudian data anggota akan diinputkan ke dalam sistem informasi dan akan diberikan kartu anggota oleh pustakawan.

- Mahasiswa yang ingin meminjam buku sebelumnya dapat melakukan pencarian buku dengan memasukkan data judul, pengarang, dan penerbit. Peminjaman mhs dapat meminjam maksimal 3 buku, untuk masa peminjaman selama 1 minggu dan transaksi peminjaman akan dimasukkan ke sistem informasi oleh pustakawan dengan membuka form pinjam, masukan id peminjam, tgl pinjam, dan data buku.
- Pengembalian buku dilakukan mahasiswa kepada pustakawan, pustakawan akan memasukkan data transaksi pengembalian dengan mengecek anggota serta buku. Jika mhs terlambat mengembalikan buku atau buku dalam keadaan rusak maka dikenakan biaya denda. Jika buku hilang, maka dilakukan penggantian buku tersebut.
- Penambahan buku baru, pustakawan dapat menambah buku baru dengan memasukkan data buku dan akan tercipta kode buku yang akan memodifikasi (update) file/table buku.