

Use Case Diagram dan Activity Diagram

Agenda



1. Pengantar Use Case Diagram



2. Use Case Detail



3. Pengantar Activity Diagram

1. Pengantar Use Case Diagram

Use Case Diagram (1)

- Menceritakan apa yang sistem akan lakukan (*what the system will do*).
- Menekankan “apa” yang diperbuat sistem, dan bukan “bagaimana”.
- Menggambarkan fungsionalitas yang diharapkan dari sebuah sistem.
- Menggambarkan kebutuhan sistem dari sudut pandang pengguna (*user*).

Use Case Diagram (2)

- Merepresentasikan sebuah interaksi antara aktor dengan sistem.
- ***Use case*** merupakan sebuah pekerjaan tertentu, misalnya : Melakukan login ke sistem, Menambah data barang baru, Mencetak laporan penjualan, dan sebagainya.
- Seorang/sebuah ***actor*** adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

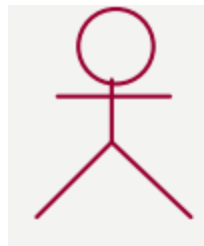
Manfaat *Use Case Diagram* (1)

- Menjelaskan *requirements*.
- Memudahkan komunikasi dengan *client*.
- *Client* mudah memahami dan memberikan masukan.
- Mempermudah pembuatan *test case* pada tahap pengujian *software*.

Komponen *Use Case Diagram* (1)

- **Actor**

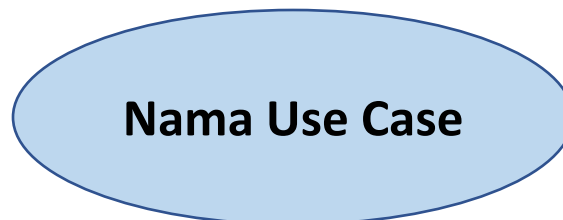
- Adalah seseorang atau apa saja yang berhubungan dengan sistem yang sedang dibangun.
- Sebaiknya diberi nama dengan kata benda.
- Dalam UML direpresentasikan dengan notasi berikut ini :



Komponen *Use Case Diagram* (2)

- ***Use case***

- Adalah bagian fungsionalitas tingkat tinggi yang disediakan oleh sistem.
- Menggambarkan bagaimana seseorang menggunakan sistem (berinteraksi dengan sistem).
- Nama *use case* selalu diawali dengan kata kerja aktif.
- Dalam UML direpresentasikan dengan notasi berikut ini :



Komponen *Use Case Diagram* (3)

- ***Relasi***

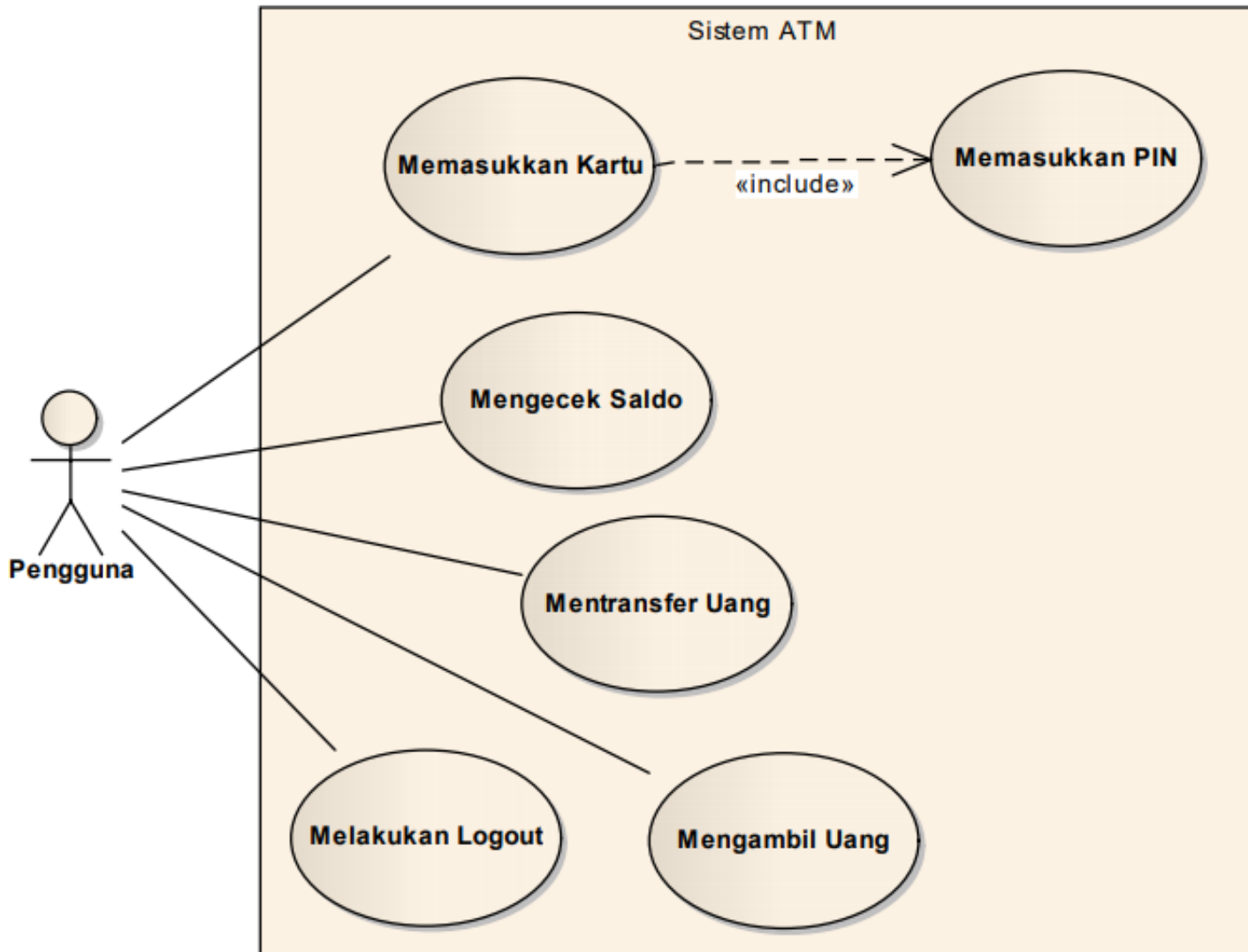
- Menghubungkan antara *actor* dan *use case*.
 - Ada 4 jenis relasi :
 - Association
 - Generalization
 - Include
 - Extend
 - Dalam UML direpresentasikan dengan notasi berikut ini :
-

Komponen *Use Case Diagram* (4)

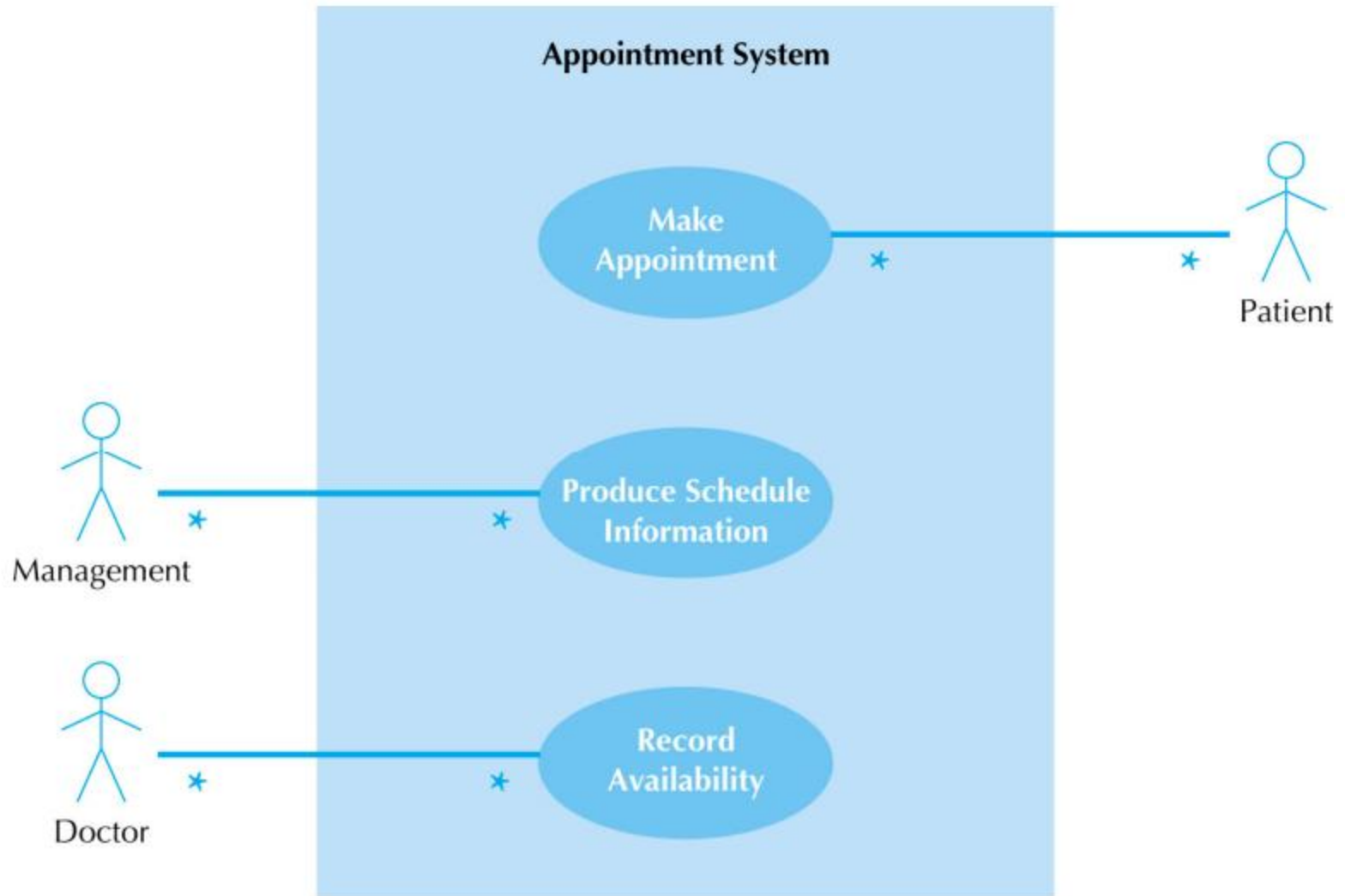
- ***System boundary boxes***

- Ditulis dengan nama *software* / aplikasi.
- Menampilkan batasan sistem dalam bentuk kotak.
- *Actor* berada di luar kotak.

Contoh Use Case Diagram (1)



Contoh Use Case Diagram (2)

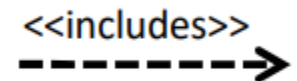


Jenis Relasi

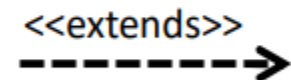
Association Relationship



Include Relationship



Extend Relationship



Generalization Relationship



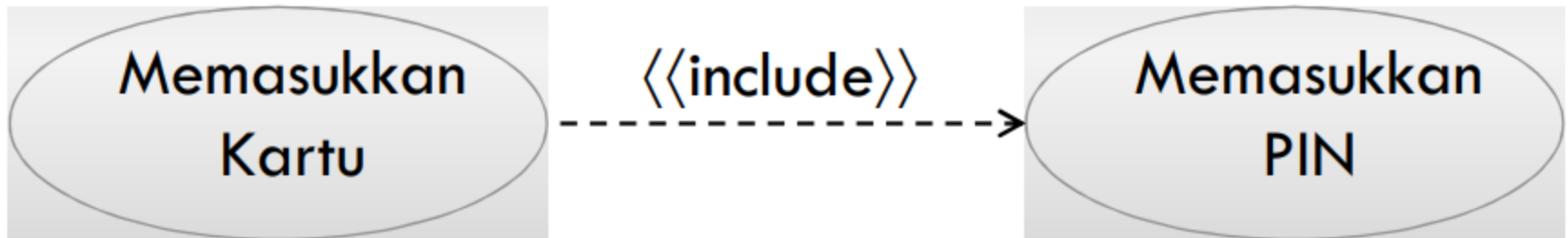
Association Relationship

- Link antar *actor* dan *use case*.
- Dapat menggunakan tanda (*) untuk menunjukkan "multiplicity of the Association".



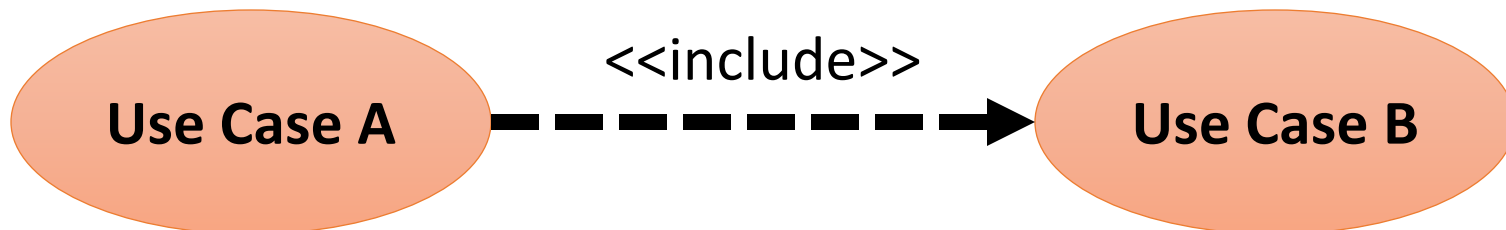
Include Relationship (1)

- Pemanggilan / penyertaan *use case* lain yang wajib / diharuskan (*required*) pada sistem.
- Tanda panah terbuka mengarah ke *sub use case*.



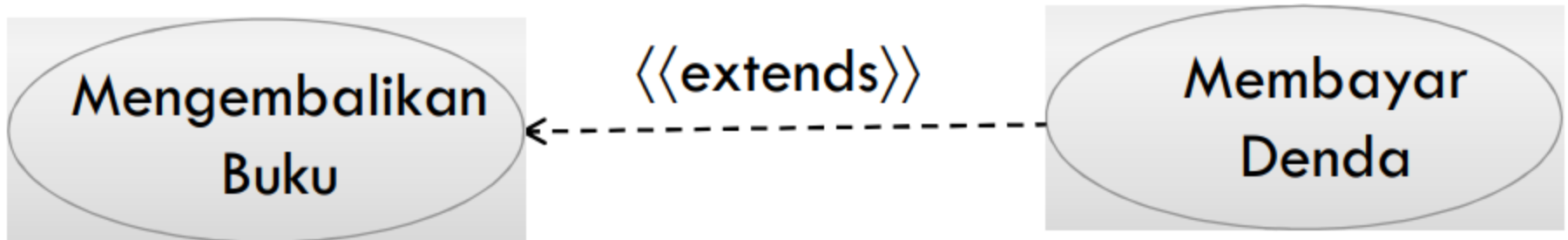
Include Relationship (2)

- Jika Use Case A <<include>> Use Case B, artinya :
Use Case B pasti dijalankan ketika Use Case A dieksekusi (dijalankan).
- Notasi :



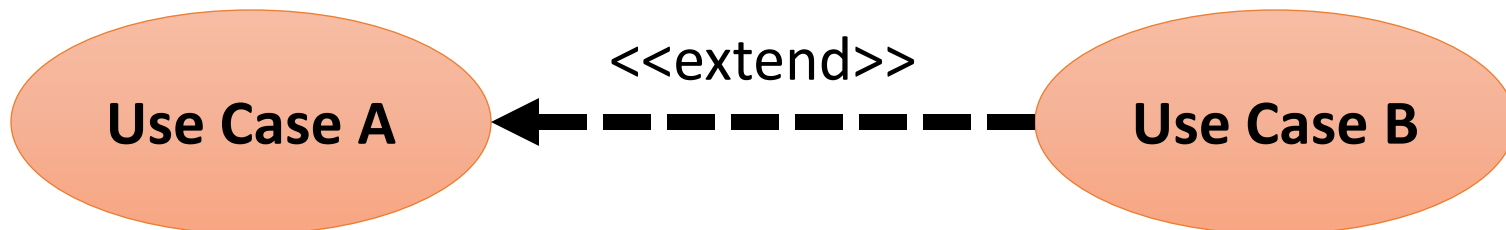
Extend Relationship (1)

- Perluasan dari *use case* jika terdapat kondisi atau syarat yang mungkin terpenuhi.
- Memperluas *use case* untuk memasukkan perilaku **opsional**.
- Tanda panah terbuka mengarah ke *parent/base use case*.



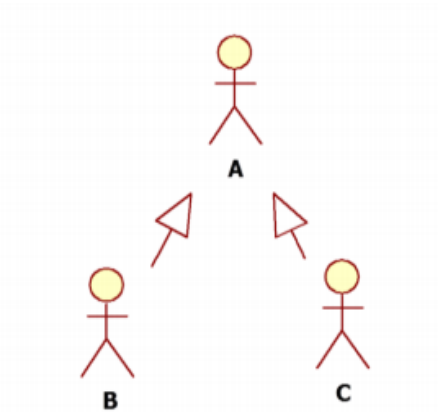
Extend Relationship (2)

- Jika Use Case B <<extend>> Use Case A, artinya :
Use Case B belum tentu dijalankan ketika Use Case A dieksekusi (dijalankan).
- Notasi :

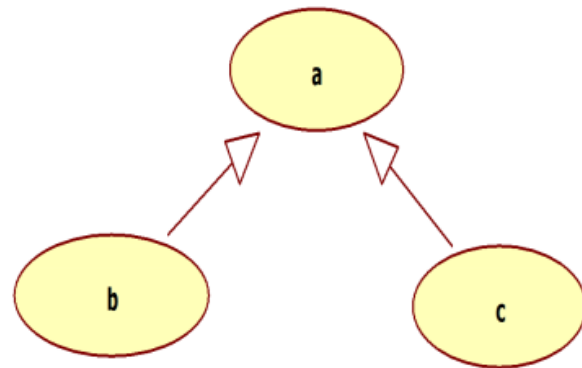


Generalization Relationship

- Digambarkan dengan sebuah garis berpanah tertutup pada salah satu ujungnya yang menunjukkan lebih umum.
- Dapat digunakan pada *use case* dan *actor*.

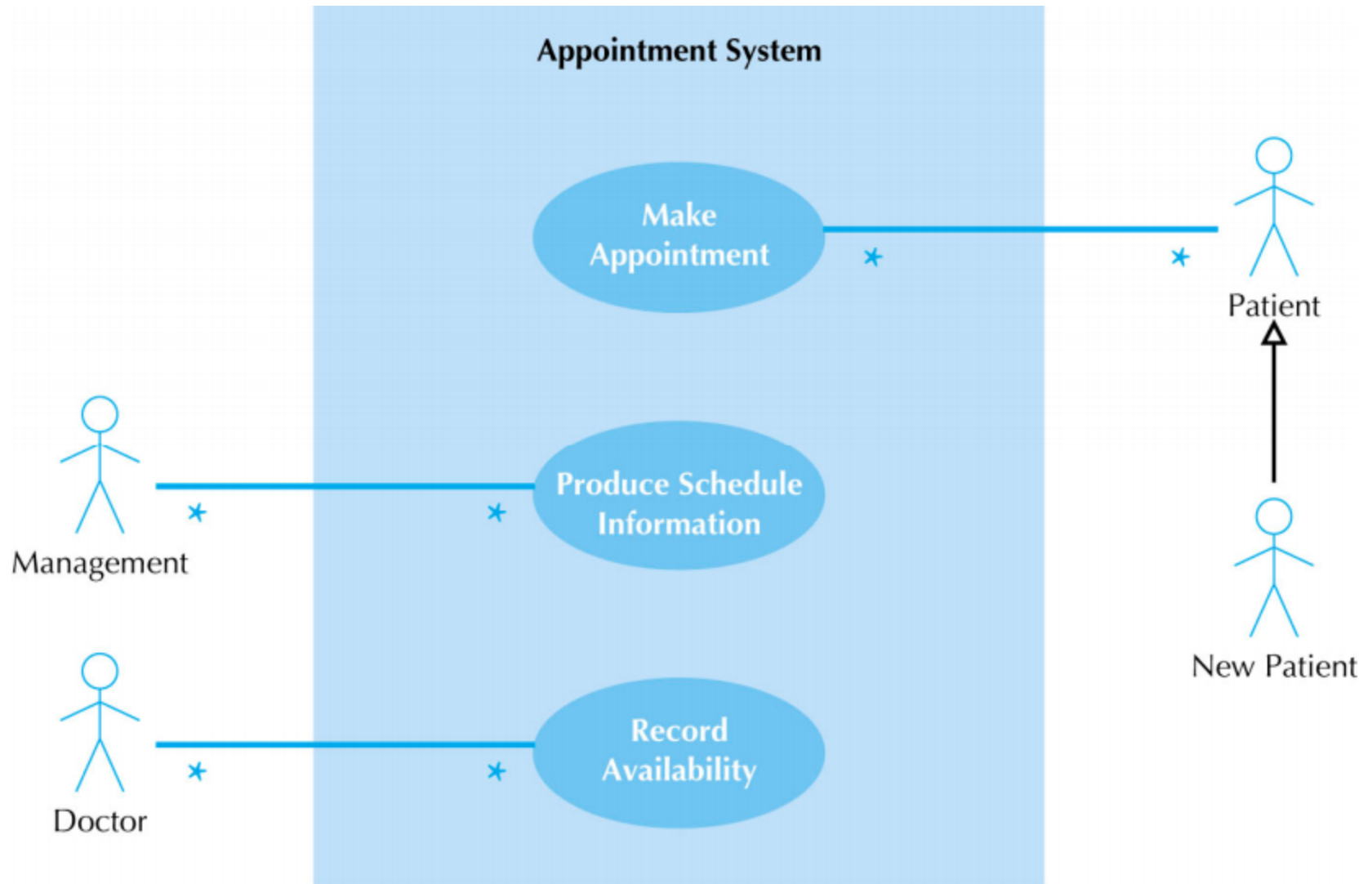


Actor Generalization



Use Case Generalization

Contoh *Use Case Diagram* Dengan *Generalization* Pada Actor



2. Use Case Detail

Use Case Detail

- *Use case name*
- *(Primary) actor*
- *Description*
- *Trigger*
- *Precondition*
- *Postcondition*
- *Scenario*

Contoh : Aplikasi Perpustakaan



Contoh *Use Case Detail*

- ***Use case name*** : Meminjam Buku
- ***Actor*** : Anggota Perpustakaan
- ***Description*** : Anggota perpustakaan meminjam buku dengan menunjukkan kartu anggota.
- ***Trigger*** : Memilih menu “Meminjam Buku”.
- ***Precondition*** : Nomor Anggota Perpustakaan, Status Peminjaman
- ***Postcondition*** : Koleksi Yang Dipinjam, Status Peminjaman
- ***Scenario*** : Anggota perpustakaan menyerahkan kartu anggota dan buku yang akan dipinjam ke petugas perpustakaan. Petugas perpustakaan akan mengecek dan memproses peminjaman buku. Jika sudah mencapai batas jumlah buku yang dipinjam, maka tidak dapat diproses.

3. Pengantar Activity Diagram

Activity Diagram (1)

- Menggambarkan aliran fungsionalitas dalam suatu sistem informasi.
- Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses.
- Secara lengkap, *activity diagram* mendefinisikan dimana *workflow* dimulai, dimana berakhirnya, aktifitas apa yang terjadi selama workflow, dan bagaimana urutan kejadian aktivitas tersebut.
- Juga menyediakan pendekatan untuk proses pemodelan paralel.




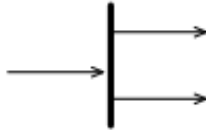
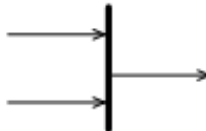

Activity Diagram (2)

- Struktur diagram ini mirip flowchart atau *Data Flow Diagram* pada perancangan terstruktur.
- Biasanya dibuat berdasarkan sebuah *use case* pada *use case diagram*.
- Berhubungan dengan *statechart diagram*.
- Diagram ini menunjukkan bagaimana aktivitas-aktivitas tersebut bergantung satu sama lain.
- Menggambarkan aktivitas sistem dan bukan apa yang dilakukan *actor*.

Activity Diagram (3)

- Terdapat “*swimlane*” yang menandakan actor yang terlibat dalam Activity Diagram.
- Memiliki tanda “start” dan “end”.

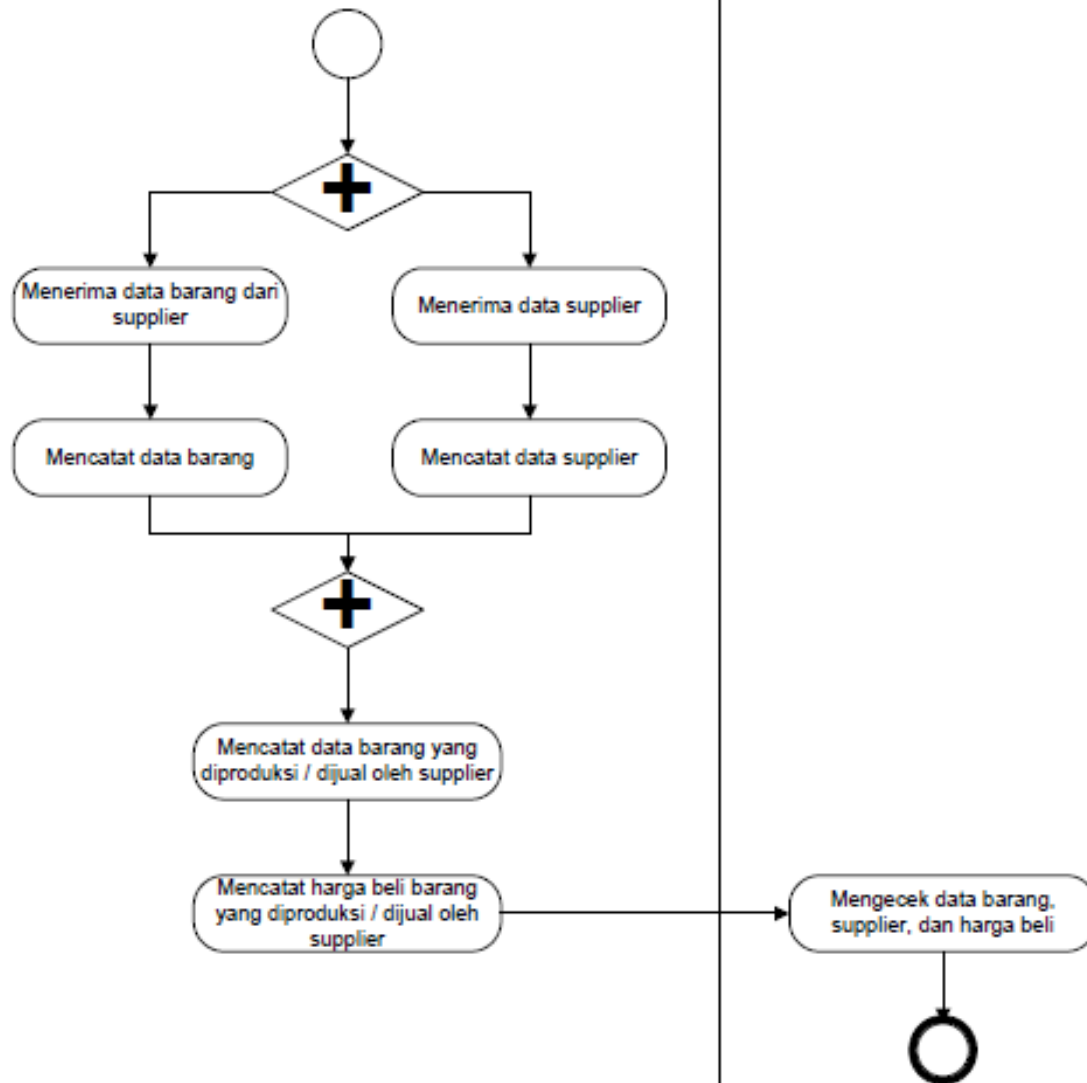
Simbol / Notasi Pada *Activity Diagram*

Simbol	Keterangan
	Start Point
	End Point
	Activities
	Fork (Percabangan)
	Join (Penggabungan)
	Decision
Swimlane	Sebuah cara untuk mengelompokkan activity berdasarkan Actor (mengelompokkan activity dalam sebuah urutan yang sama)

Pencatatan dan Pembaruan Data Barang & Supplier

Staf Div. Inventori

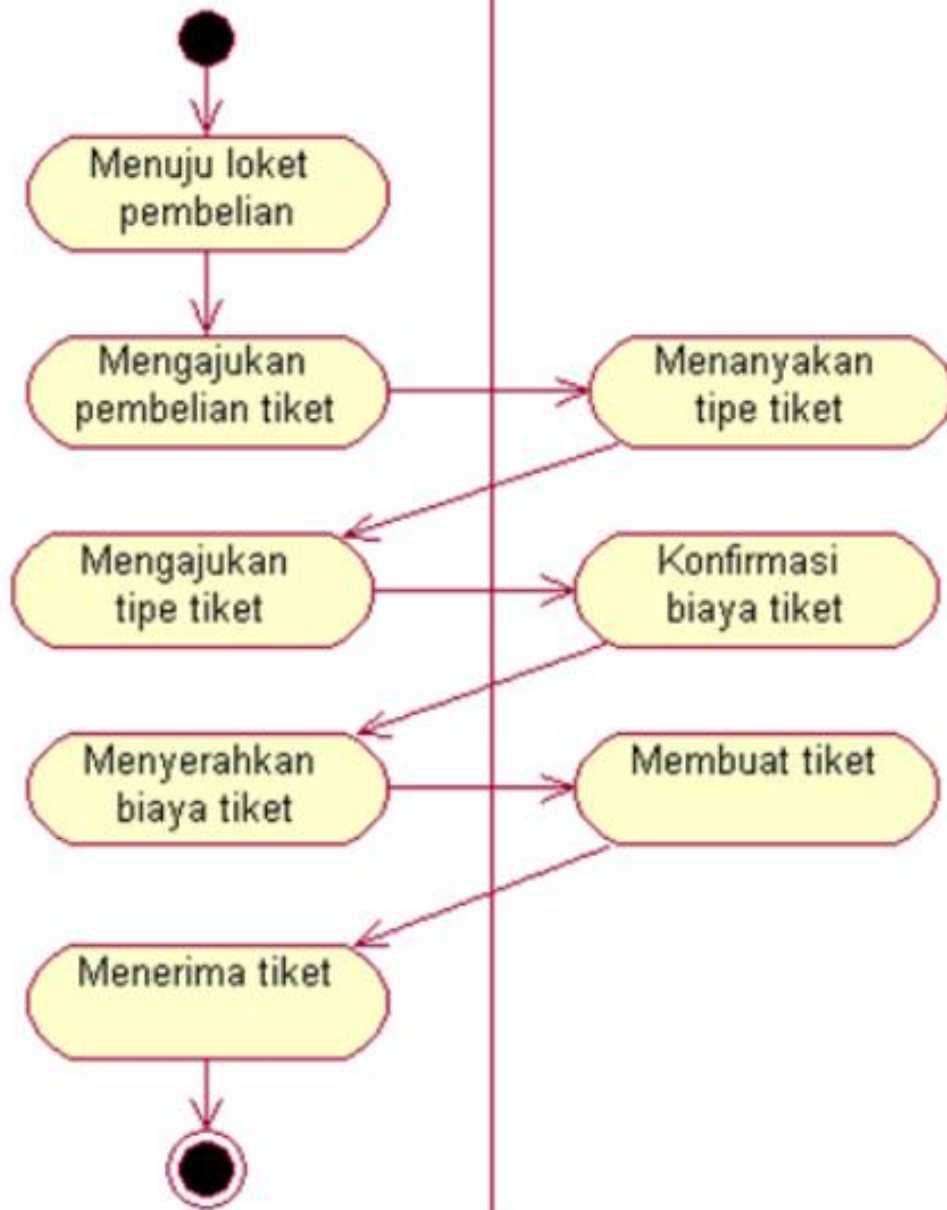
Manajer Div. Inventori



Activity
Diagram
“Pencatatan
dan
Pembaruan
Data Barang
& Supplier”

: Pelanggan

: Petugas



Activity Diagram
“Pembelian tiket”

Activity Diagram Online Shopping

