

# *Analisis dan Desain Sistem Informasi*

## **OOAD: Activity Diagram**

*Team Teaching ADSI*

# Tujuan perkuliahan

- Memahami proses pemodelan kebutuhan secara berorientasi objek dengan Activity Diagram
- Mampu memodelkan kebutuhan yang dispesifikasikan dalam Use Case Model ke dalam Activity Diagram

# Topics


- Concept of process modelling
- Activity Diagram
- Process modelling using Activity Diagram

# Activity Diagram

# Definitions

- Activity diagrams represent the **dynamic aspects** of the system.
- activity diagram seperti flow chart menunjukkan **workflow dari system**.
  - Aktivitas-aktivitas dalam alur kontrol sistem,
  - Aktivitas apa yang bisa dilakukan secara parallel.
  - Jalur alteranif dalam alur sistem.
- They can show/visualize the flow across use cases or within a use case.

# Activity Diagram

- Simbol activity = 
- An activity is some task which needs to be done. Each activity can be followed by another activity (sequencing).
- Memiliki **satu start** dan **bisa banyak end**



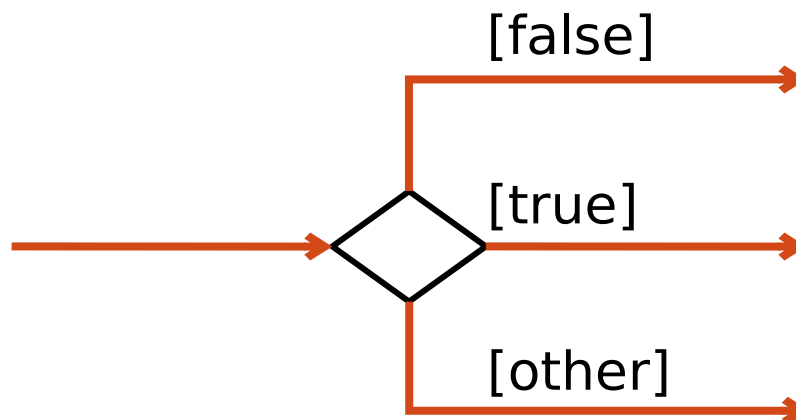
Start symbol



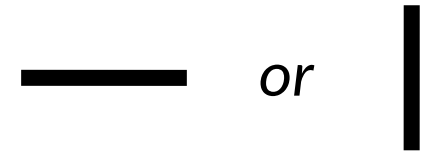
End symbol

# Decision Activities

- Represented by Diamond symbol.
- Each trigger coming from it has a guard.
- An iteration or loop is marked by \* symbol



# Parallel Process



- Represented by Synchronisation bar.
- All triggers from this attach to activities that can occur in parallel, with no specific sequence, or concurrently.
- Synchronisation bar **come in pair**.
- The next synchronisation bar closes the concurrency.



# Activity Diagram for Use Cases

- They can be used for describing either
  - Use cases scenarios or
  - Complicated methods
- You can attach activity diagrams to most model elements in the use case or logical views.
- Very effective in illustrating the workflow of various events in a use-case diagram.
- You can use activity diagrams to specify and define each event or scenario in a use-case diagram.

# Drawback

- Activity diagrams tell you what is happening, but not who does what.
- In domain modelling, this diagram type does not convey which people or departments are responsible for each activity.
- In programming, it does not convey which class is responsible for each activity.

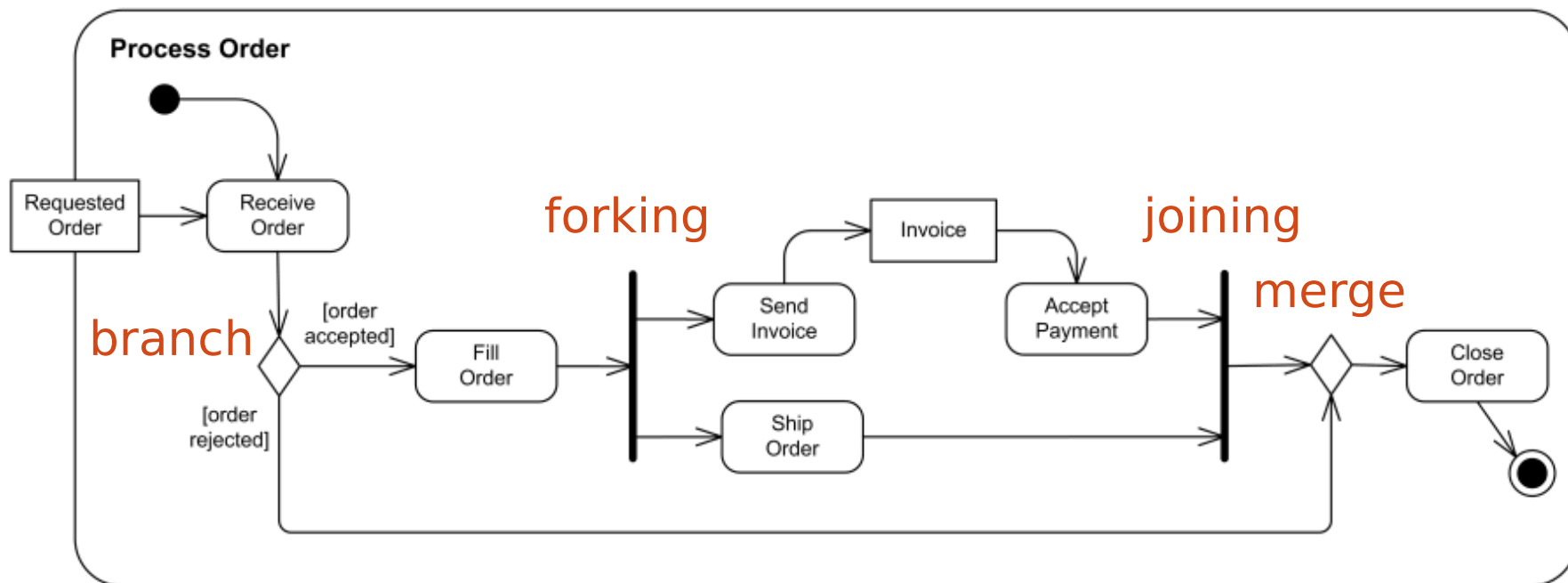
# Swimlanes

- You can put activities into swimlane to tell **who does what** or assigns class responsibilities to activities
- Arrange activity diagrams into vertical (or horizontal) zones separated by dashed lines.
- Each zone represents the responsibilities of a particular class or department.

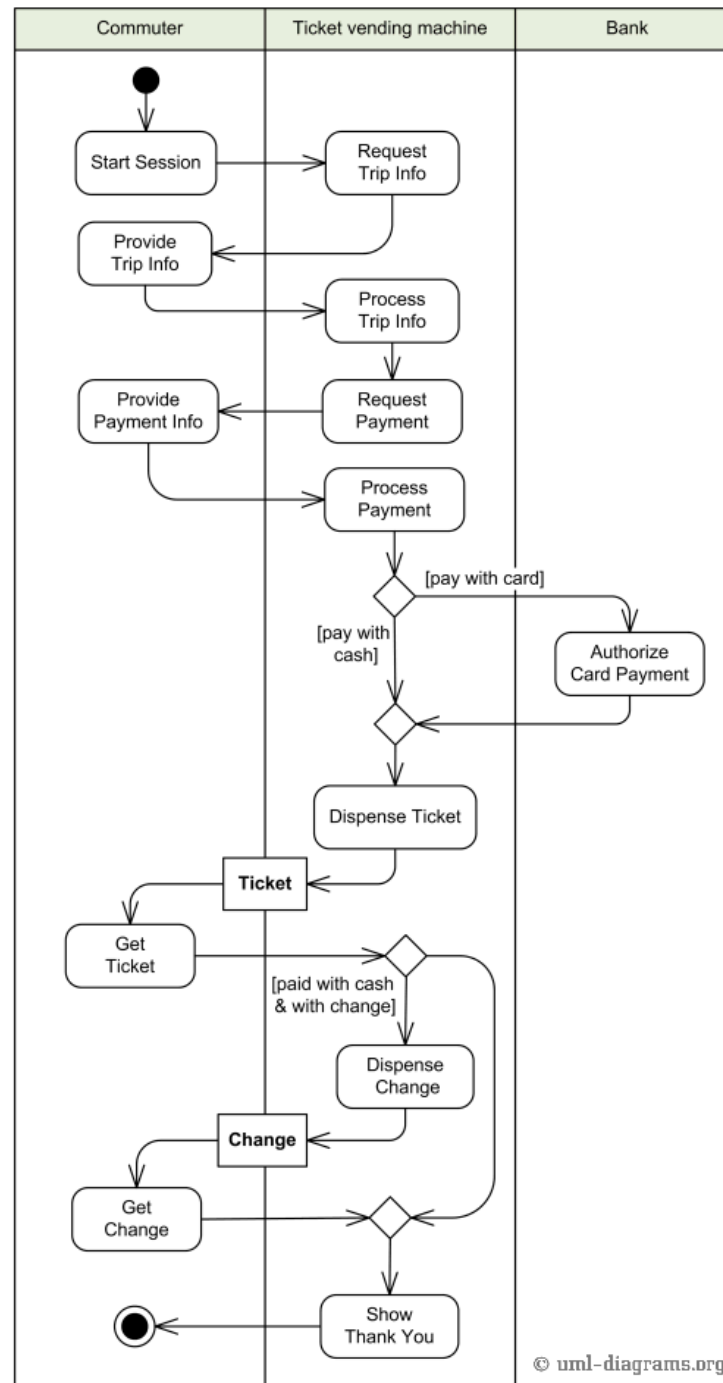
# When to Use Activity Diagrams

- Do use them for
  - Analysing Use Cases.
  - Understanding workflow across many Use Cases.
  - Dealing with multi-threaded applications.
- Don't use them
  - to see how objects collaborate.
  - to see how an object behaves over its lifetime.

# Example



# Activity Diagram Example Using Swimlanes



# Creating Activity Diagram

# Basic Steps in Creating Activity Diagram

1. Obtain a use case specification, identify activities and actors involved in its main flow scenario
2. Identify the main objective and make note of the specified pre/post condition
3. Draw all activities into the diagram in logical order
4. Identifies and draw all of specified alternate activities
5. Identifies parallel and decision/conditional activities and put relevant symbols accordingly
6. Connect all activities and objects with transition/flows
7. Set all actions, triggers and guard conditions



# Add swimlanes

- Decide who or what is responsible for performing the activities and states through swimlanes
  - Create swimlanes based on actors specified on the use case specification plus the system itself.
- Name each swimlane and place the appropriate activities and states within each swimlane.

# Evaluation and Analysis

- Check the objective
  - *What needs to take place or happen by the end of the workflow? What needs to be accomplished?*
- Have all forks been joined?
  - Every parallel/concurrent processes must be joined
- Do all alternative flows have been modelled?
  - Optional/Decisive, Business-ruled, and Error flow
- Do all actions, triggers and guard conditions match with the specified flow?
- Any *incomplete* flow in the diagram?