



# PEMODELAN SISTEM PERANGKAT LUNAK

Willy Sudiarto Raharjo  
Aditya Wikan Mahastama



**UML:  
Activity Diagram**

**3**



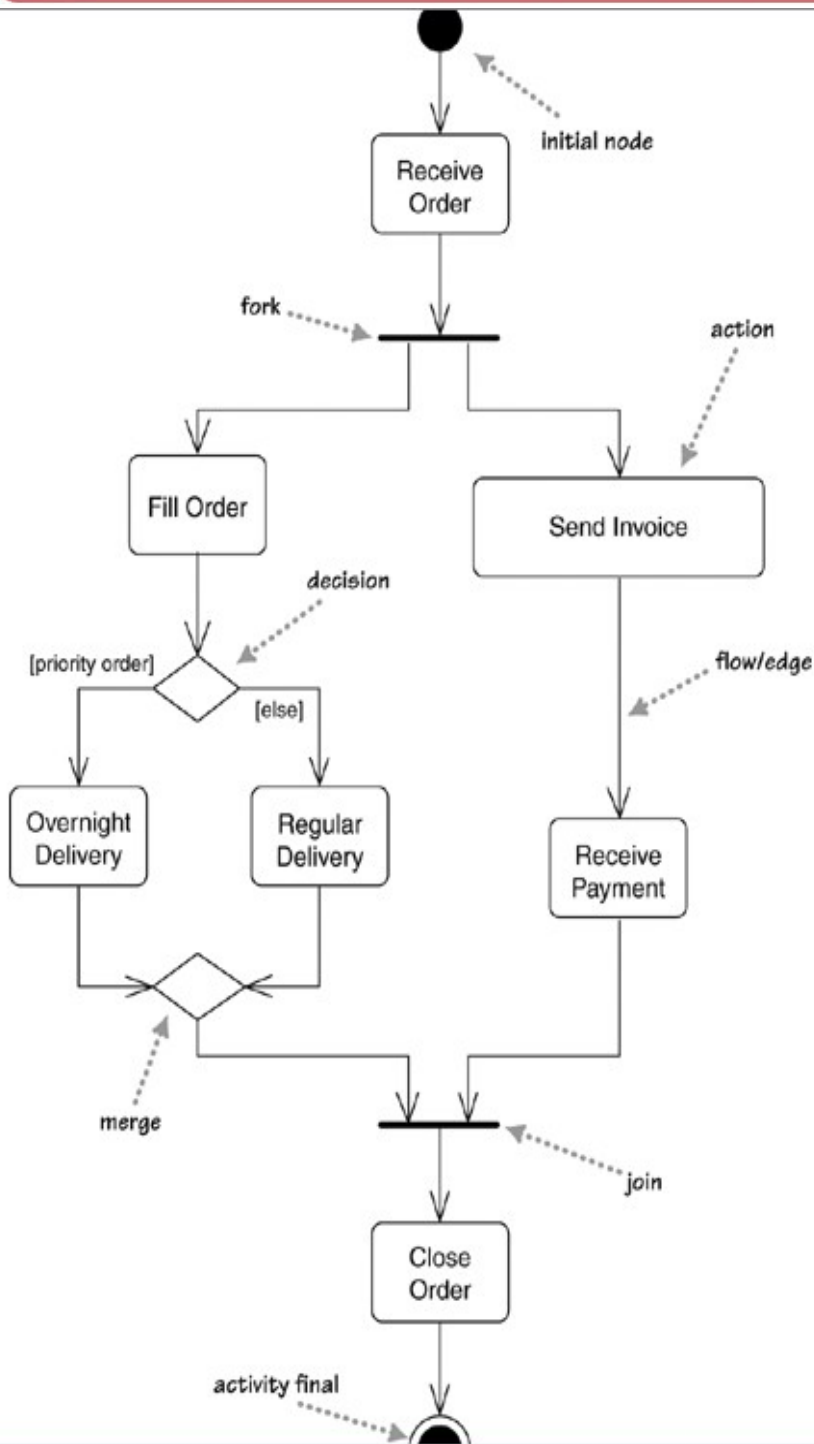
# Activity Diagram

- Teknik untuk menjelaskan business process, procedural logic, dan work flow
  - Bisa dipakai untuk menjelaskan use case text dalam notasi grafis
  - Menggunakan notasi yang mirip flow chart, meskipun terdapat sedikit perbedaan notasi



# Activity Diagram

- Nodes (initial dan final)
- Activity
- Flow/edge
- Fork
- Join
- Condition
- Decision
- Merge
- Partition
- Bisa lebih dari 1 final node
- Bisa fisik/elektronik
- Arah sebuah proses
- Awal proses paralel
- Akhir proses paralel
- Bentuk [text]
- If ... then ....
- Salah satu flow harus masuk ke merge dahulu
- Who/what performing activities

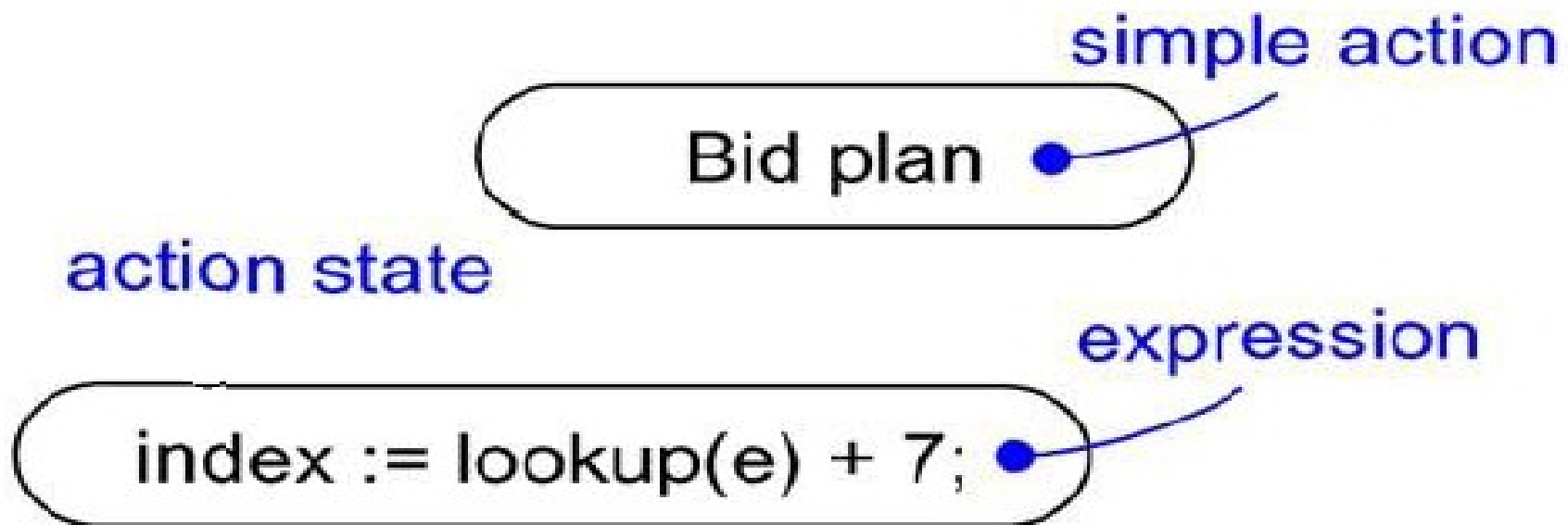


- Diawali dengan initial node
- Fill Order dan Send Invoice terjadi secara bersamaan
  - Urutan menjadi tidak relevan antara 2 proses tadi
  - Digunakan untuk concurrent algorithm atau threads
- Jika terdapat paralelism, diperlukan sinkronisasi
  - Order tidak akan ditutup sampai barang dikirim dan pembayaran diterima
  - Digunakan operasi join
- Diakhiri dengan activity final



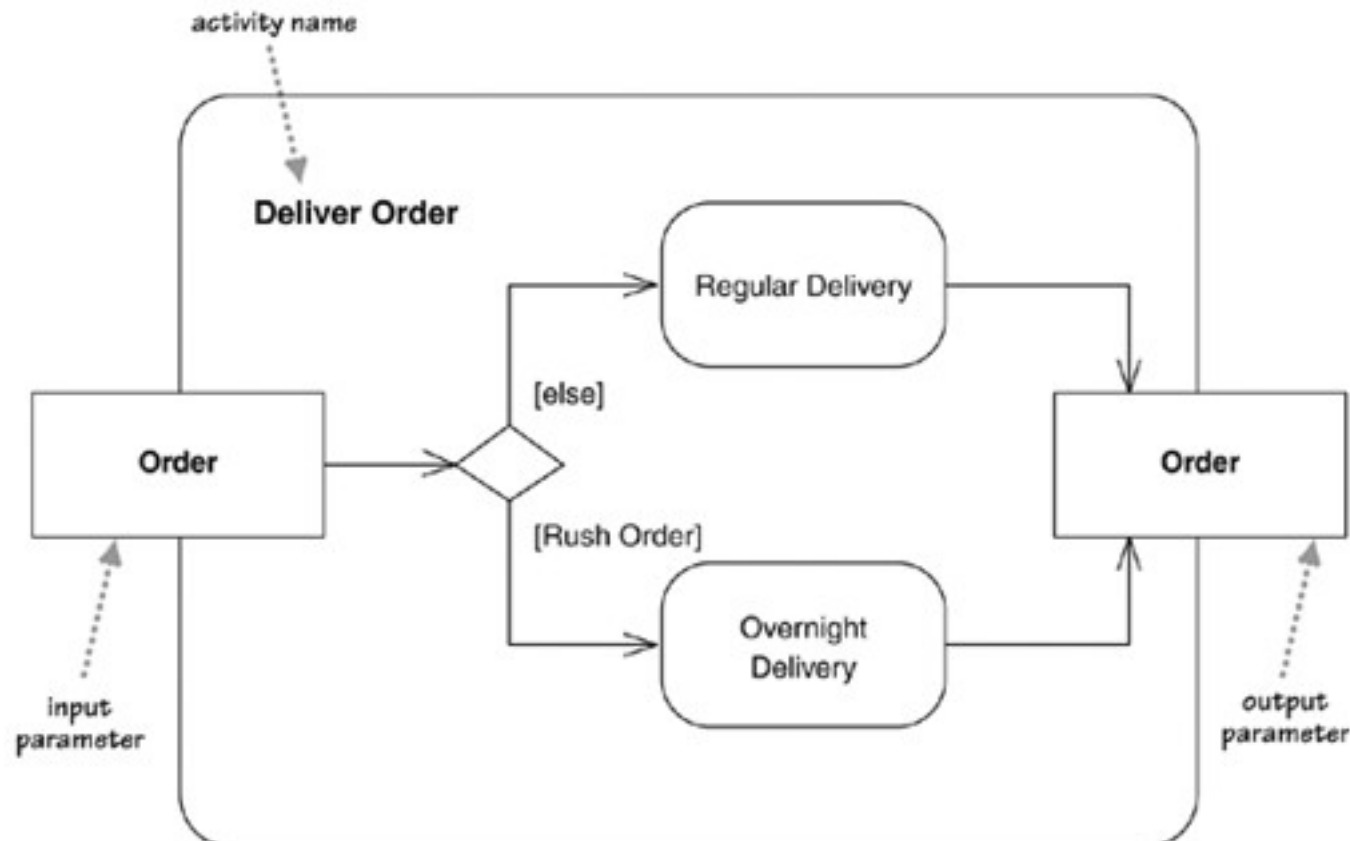
# Activity

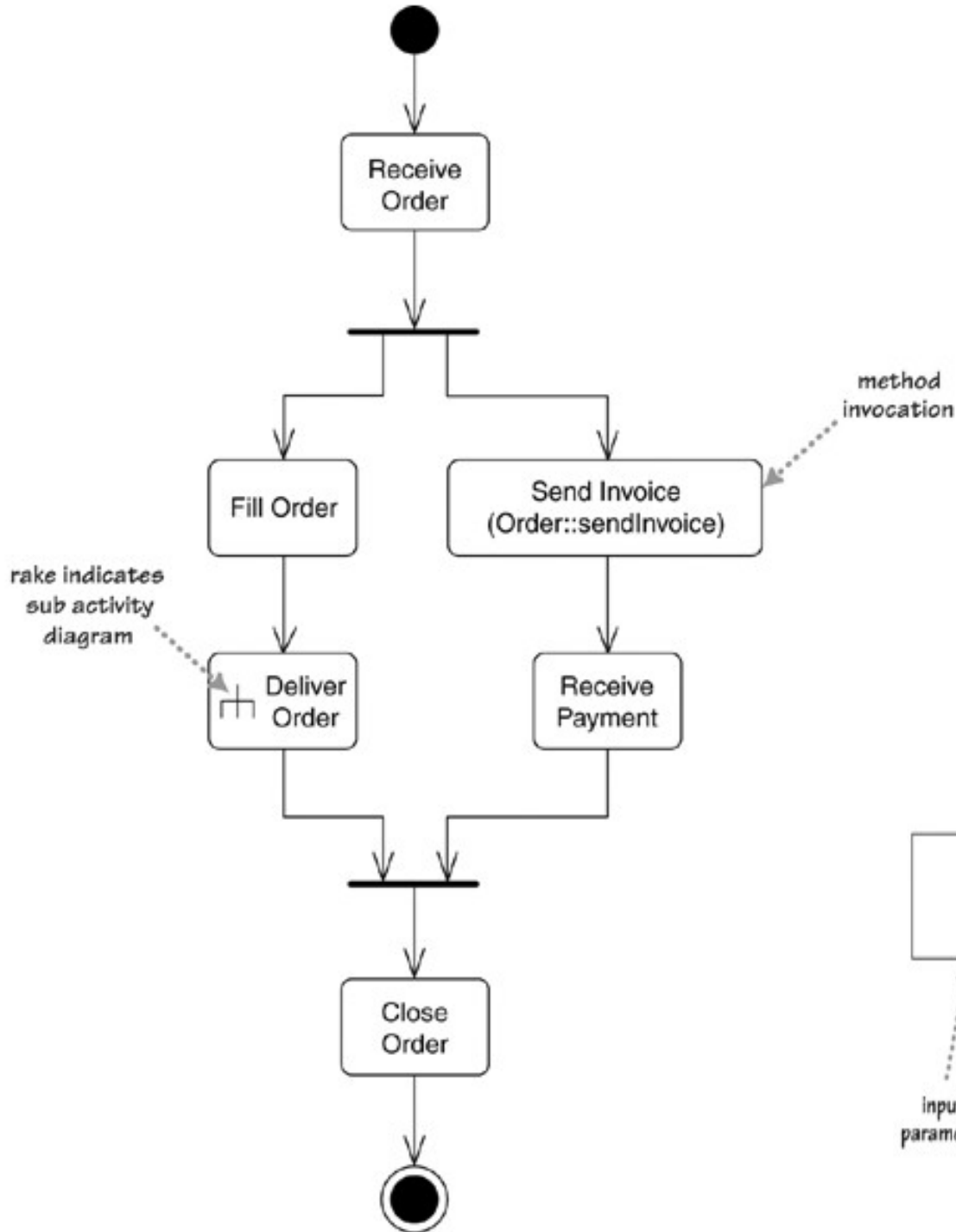
- Proses komputasi atau perubahan kondisi
- Bisa berupa kata kerja atau ekspresi
- Bersifat atomik, tidak dapat didekomposisi



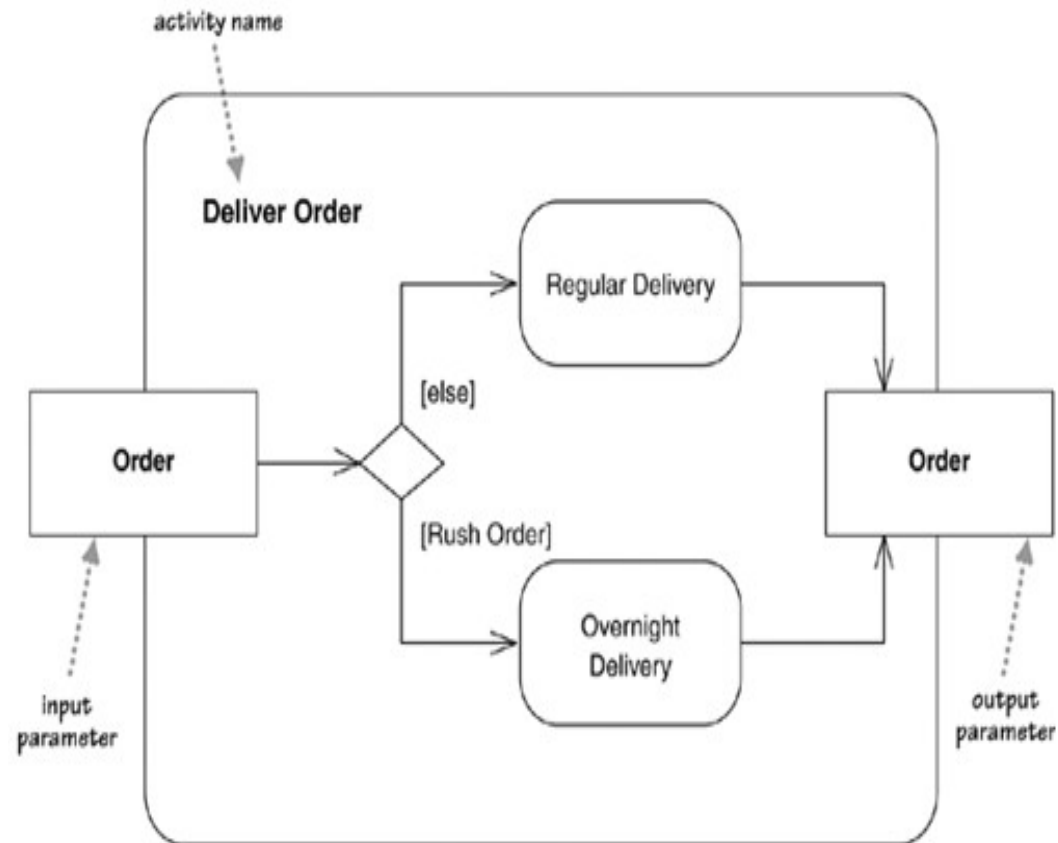
# Activity Decomposition

- Untuk menyederhanakan diagram, seringkali dilakukan dekomposisi aktivitas
- Contoh: Proses Pengiriman



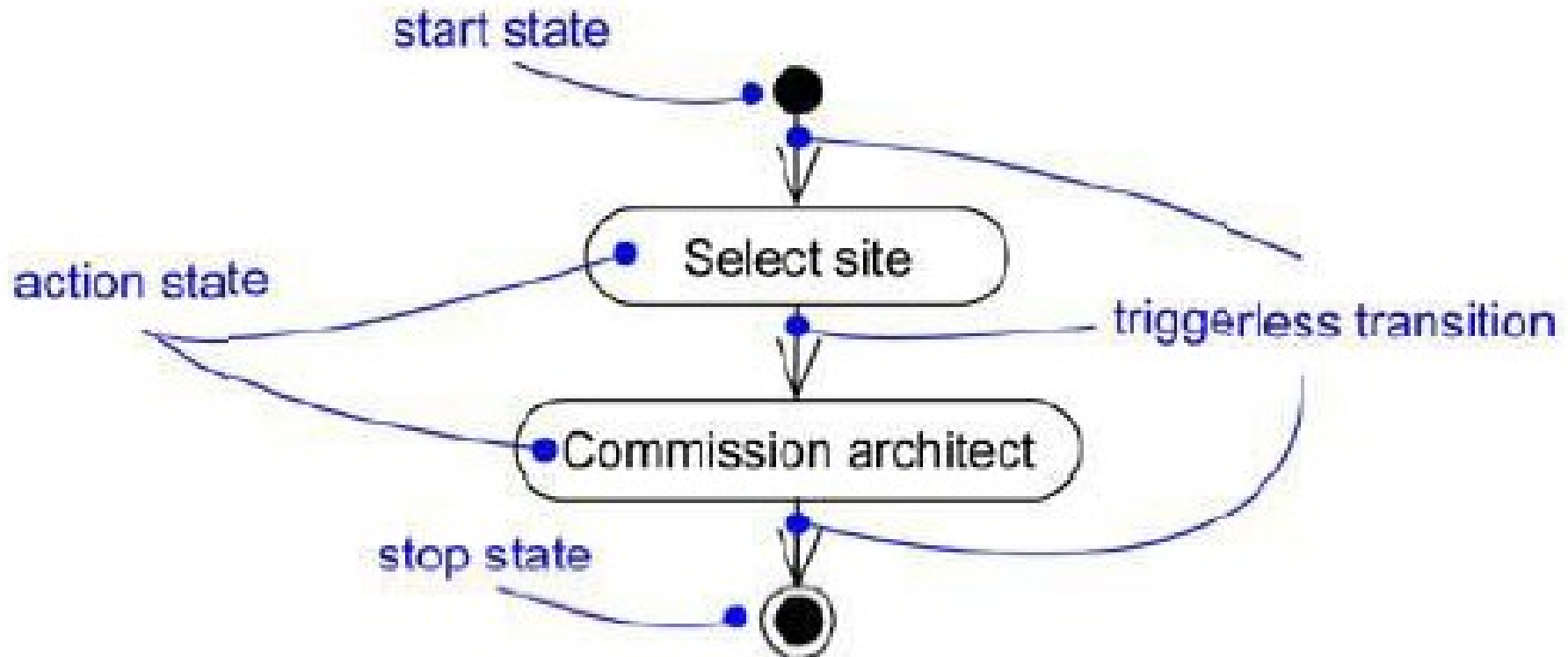


## Sub Activity: Deliver Order



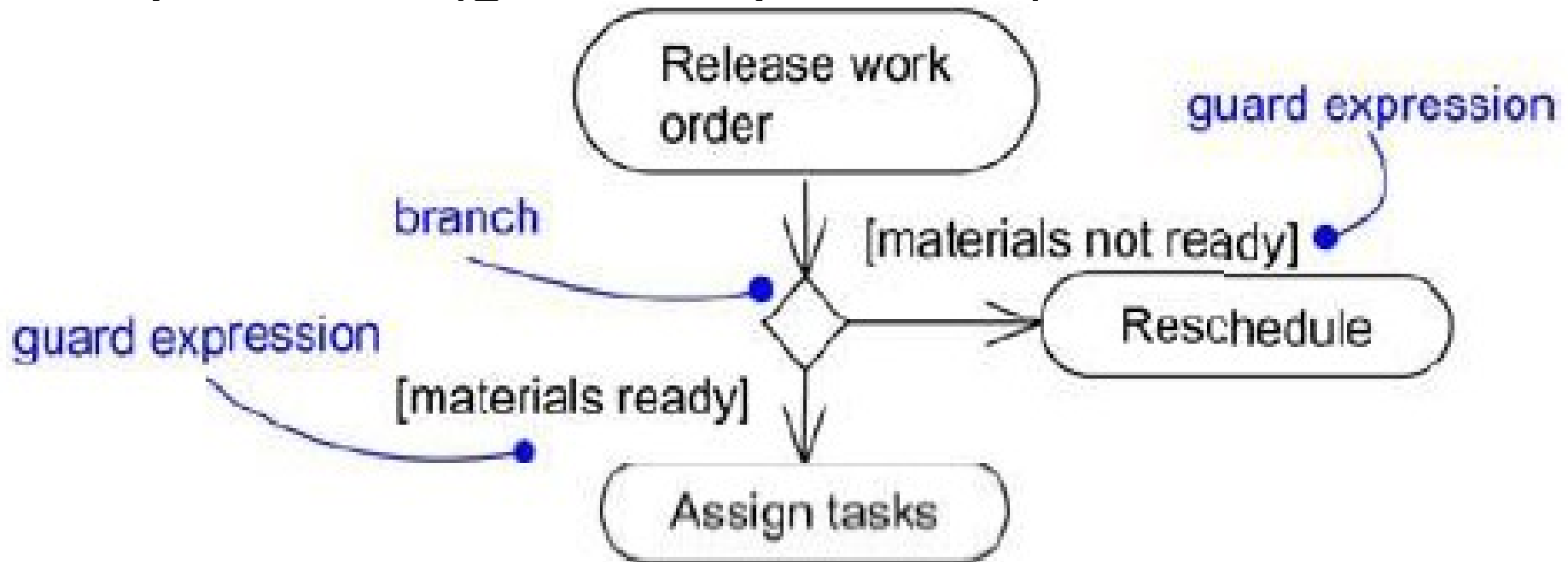
# Transitions

- Ketika sebuah aktivitas atau state selesai, maka flow control berganti ke aktivitas atau state berikutnya
- Dinotasikan dengan garis lurus dan anak panah



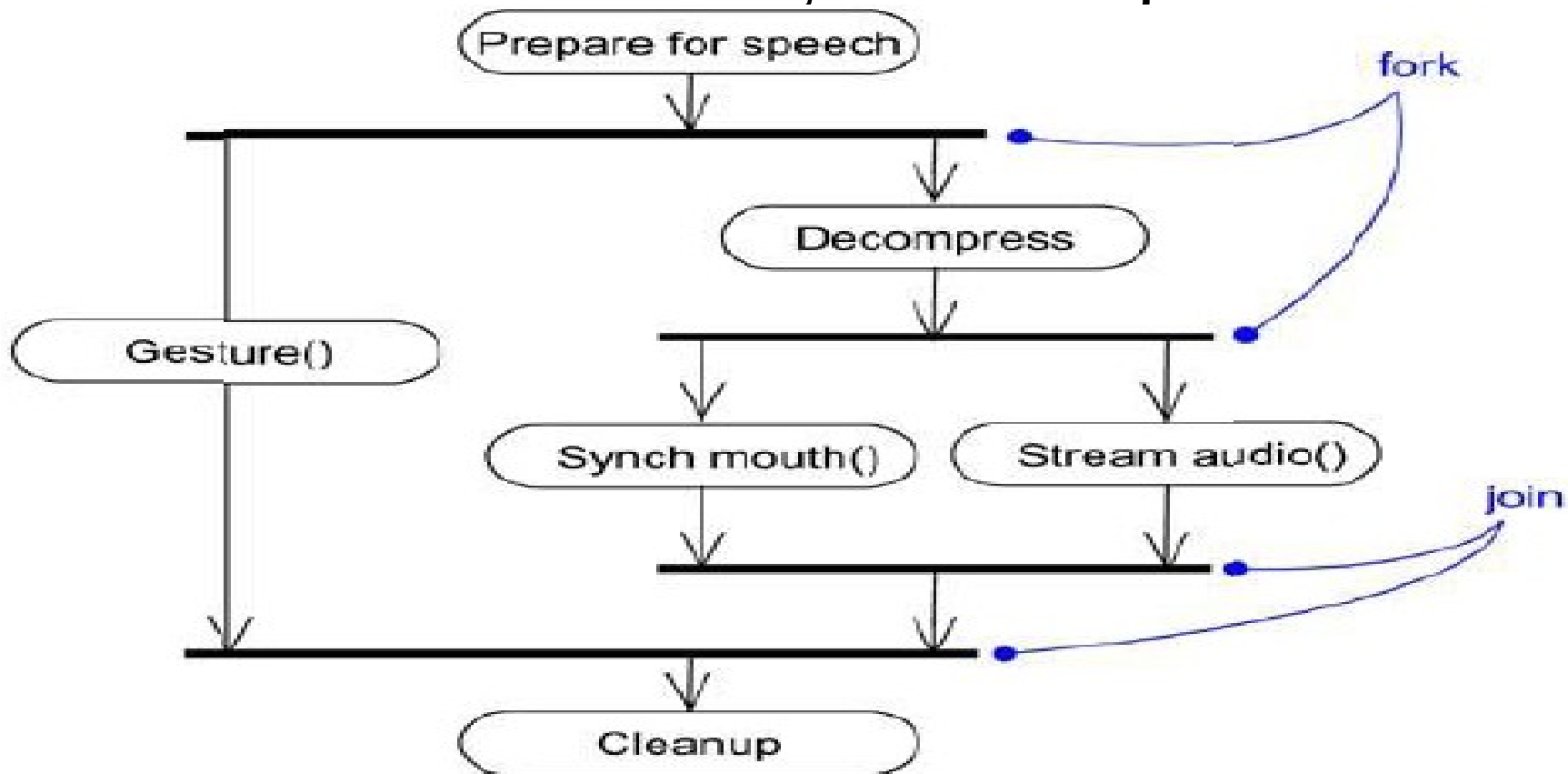
# Branching

- Jalur alternatif sesuai dengan kondisi boolean
- Dinotasikan dengan diamond
- Boleh memiliki lebih dari 3 outgoing node
- Di setiap outgoing node, tambahkan boolean expression (guard expression)



# Forking dan Joining

- Fork mampu menggambarkan aktivitas yang mungkin terjadi secara concurrent
- Dinotasikan dengan garis vertikal/horizontal
- Setelah semua selesai, harus di join kembali



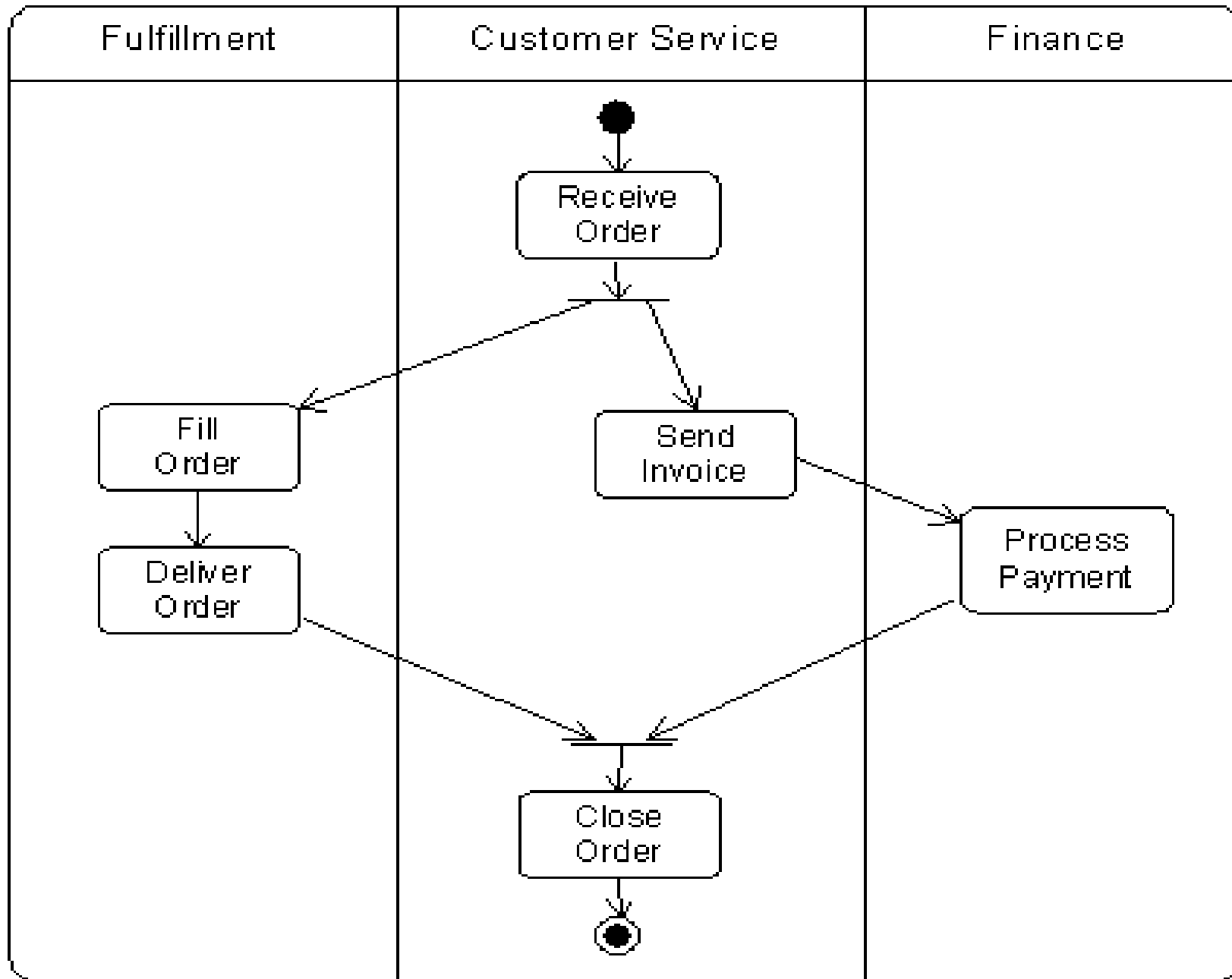


# Partitions

- Activity Diagram mampu menjelaskan apa yang terjadi, tetapi tidak mampu memberikan informasi siapa melakukan apa
- Untuk bisa melihat siapa yang bertugas melakukan sebuah proses, activity diagram bisa dipartisi
  - Swim Lane
  - Node-based

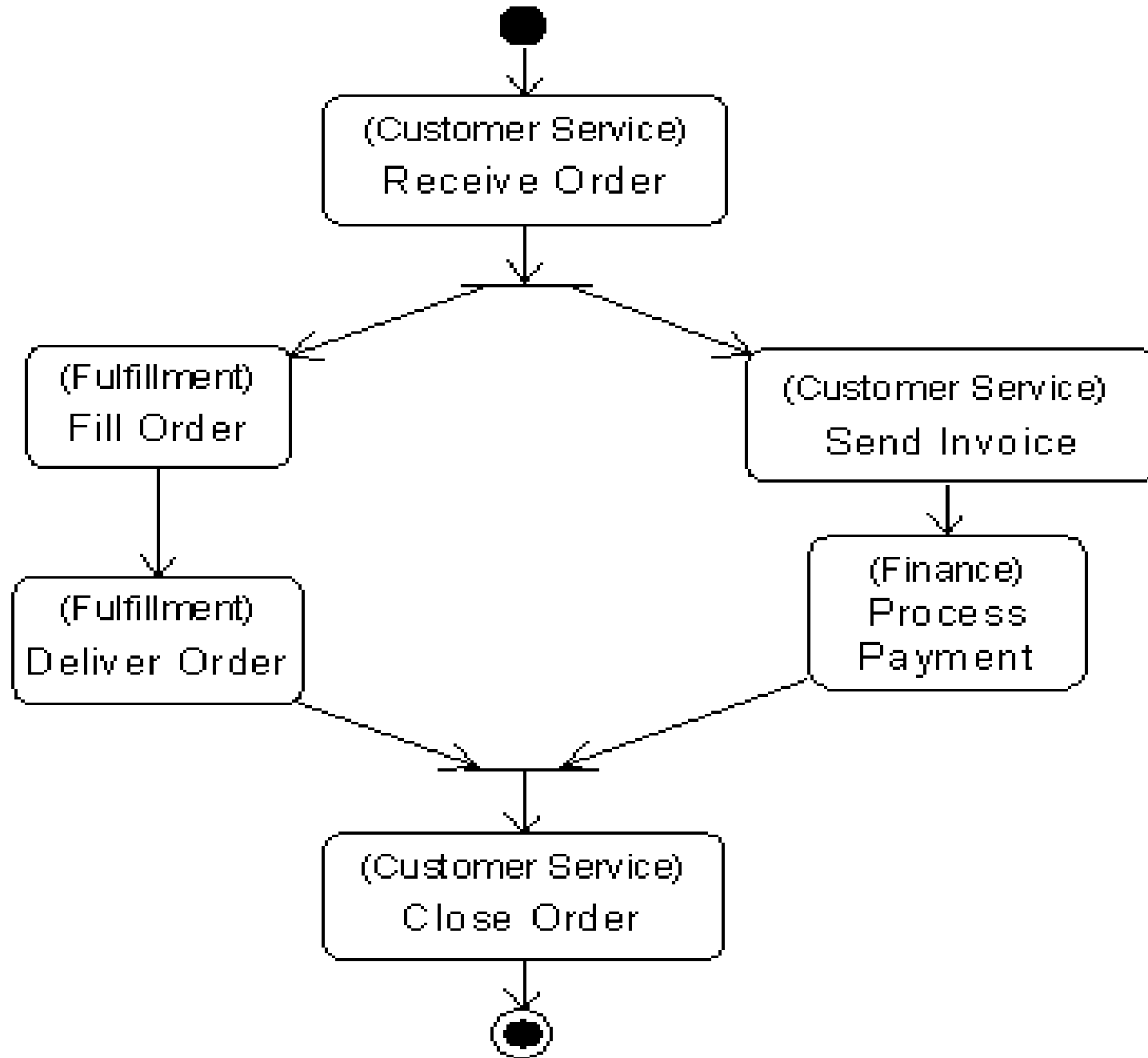


# Swim lane



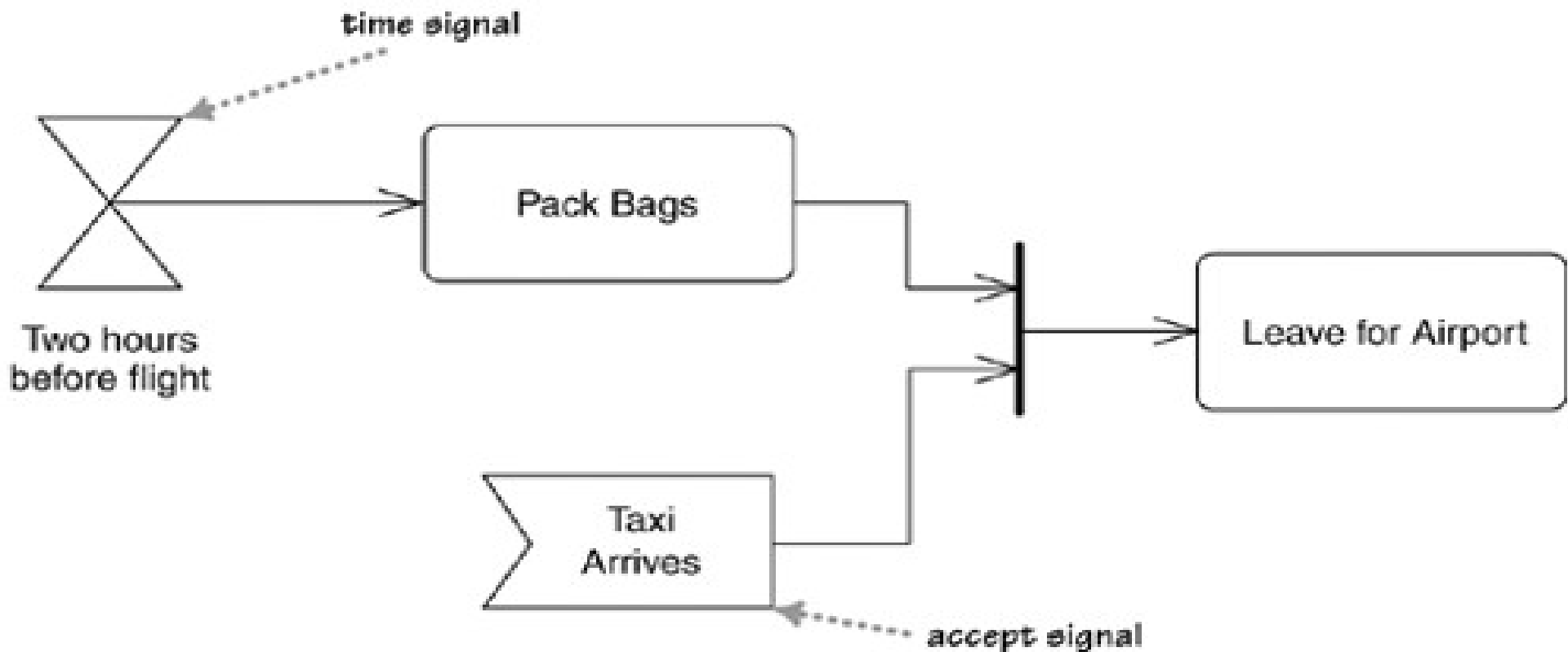


# Node Based



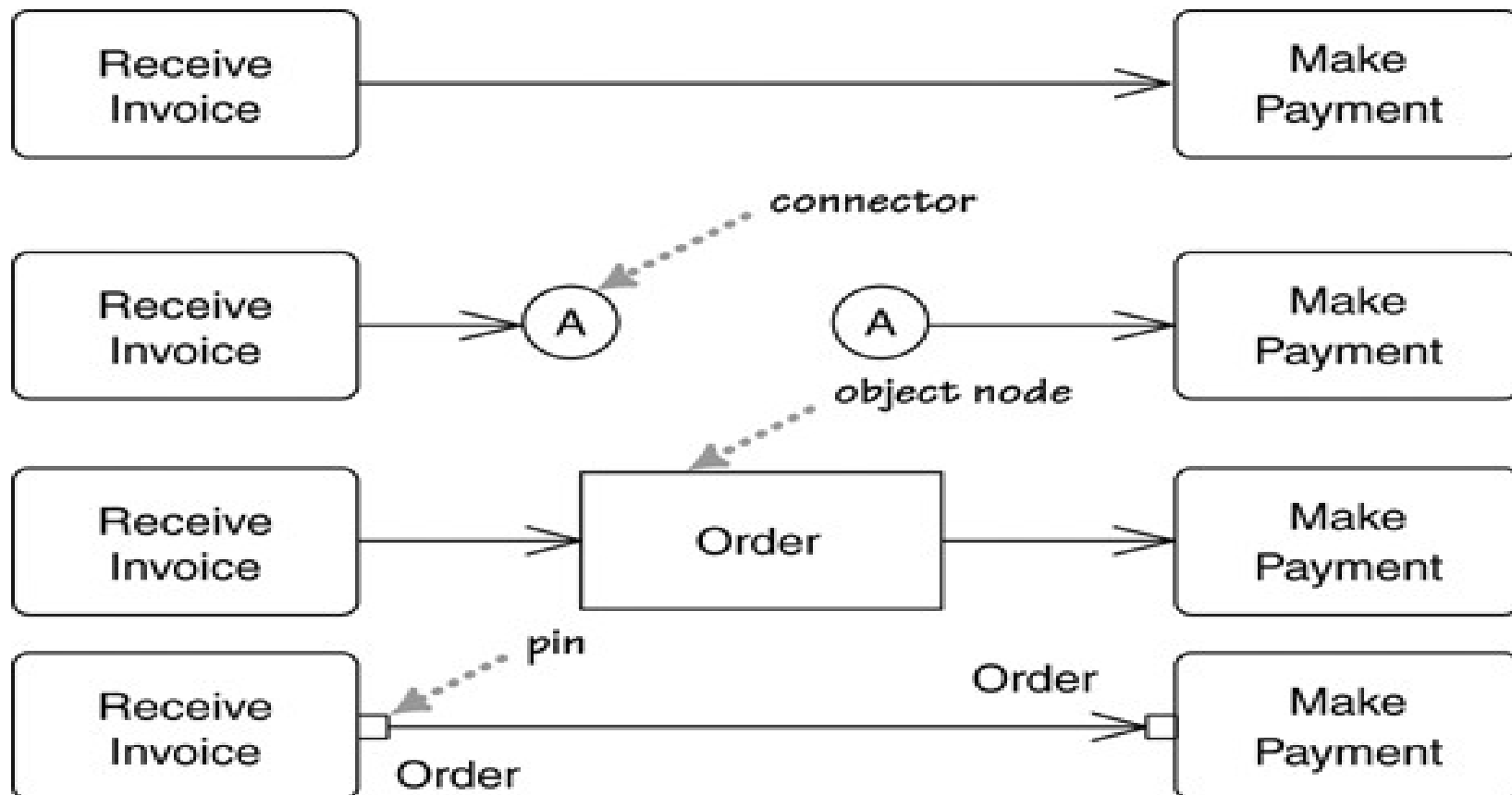
# Signal

- Tanda untuk memulai sebuah aktivitas tidak harus melalui pemanggilan fungsi, tetapi juga bisa melalui sinyal, misalnya sinyal waktu atau kejadian lain yang bisa menjadi trigger



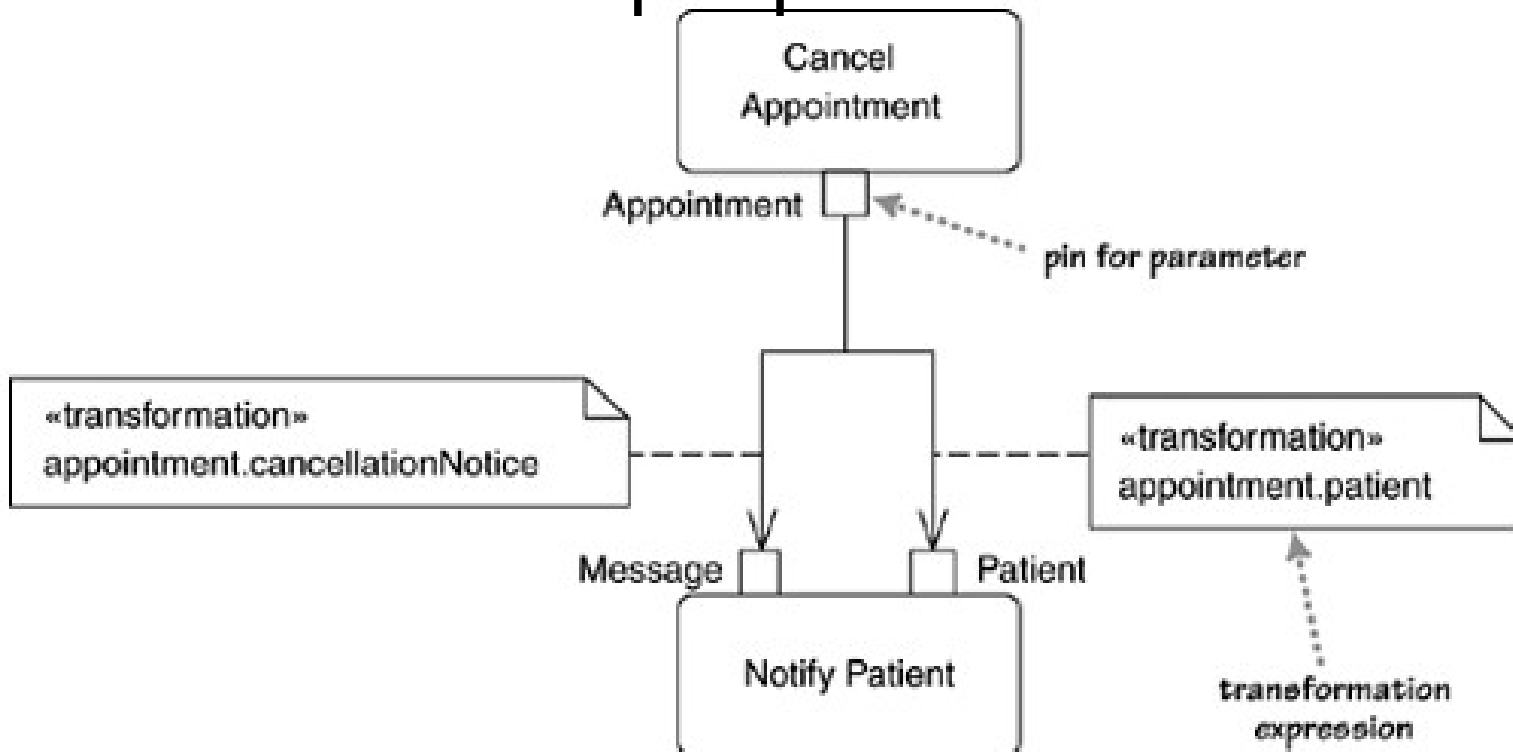
# Flow dan Edge

- UML 2 menggunakan istilah flow dan edge untuk menjelaskan hubungan antara 2 aktivitas
- Terdapat 4 cara penggambaran



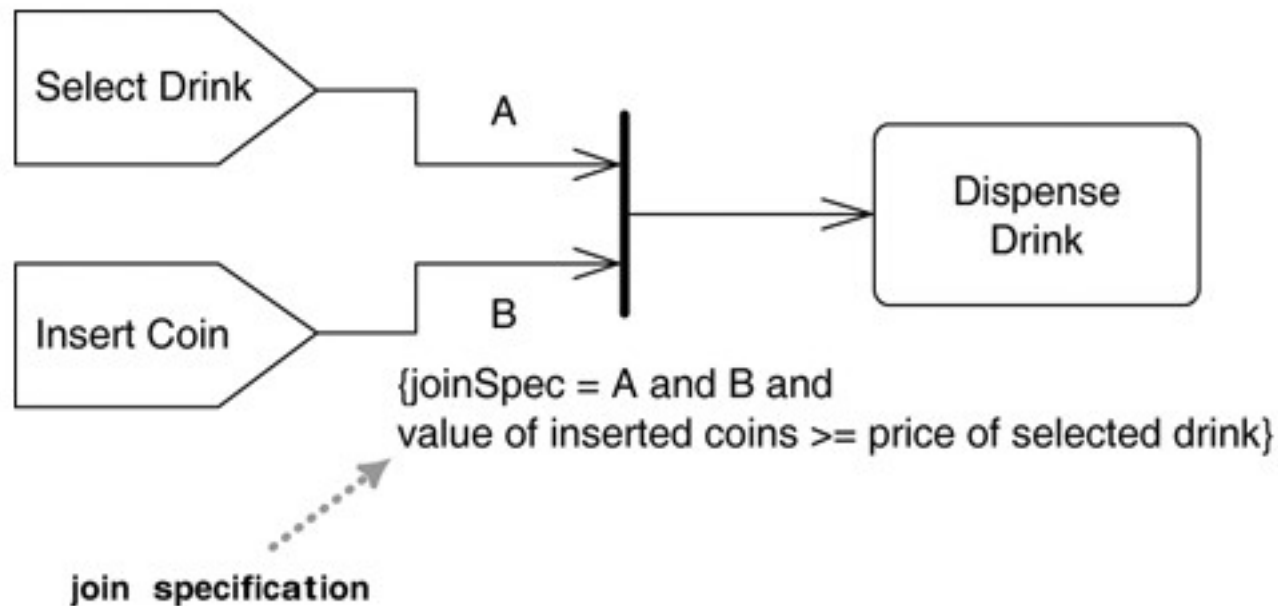
# Pins dan Transformasi

- Seringkali sebuah aktivitas memerlukan parameter, seperti pada fungsi
- Jika ingin ditampilkan, bisa menggunakan pin
- Transformasi digunakan untuk konversi input parameter dan output parameter



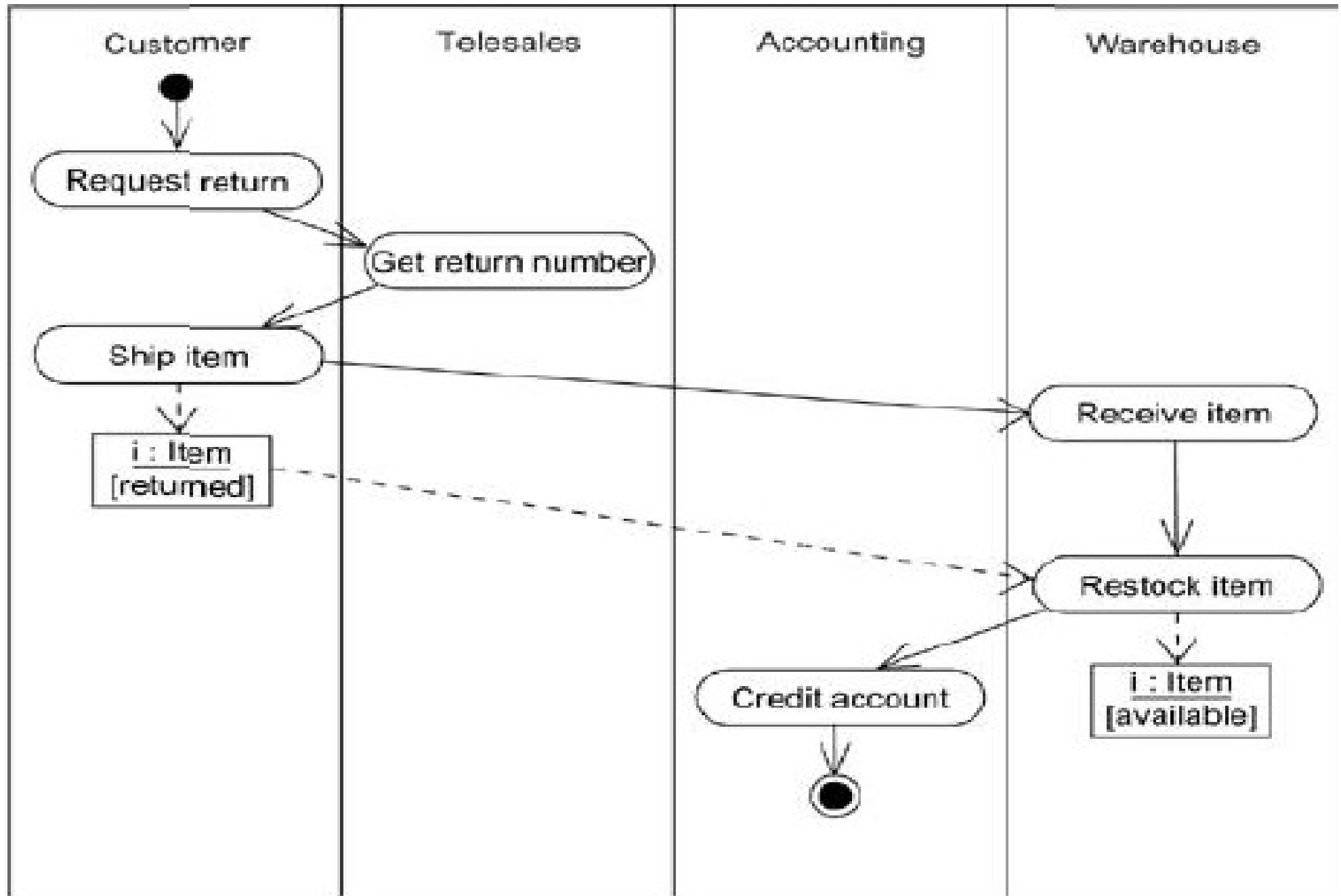
# Join Specification

- Untuk memastikan bahwa dua atau lebih proses sudah selesai sebelum melanjutkan ke proses selanjutnya, digunakan join specification
- Notasinya adalah semacam pipa





# Contoh Workflow



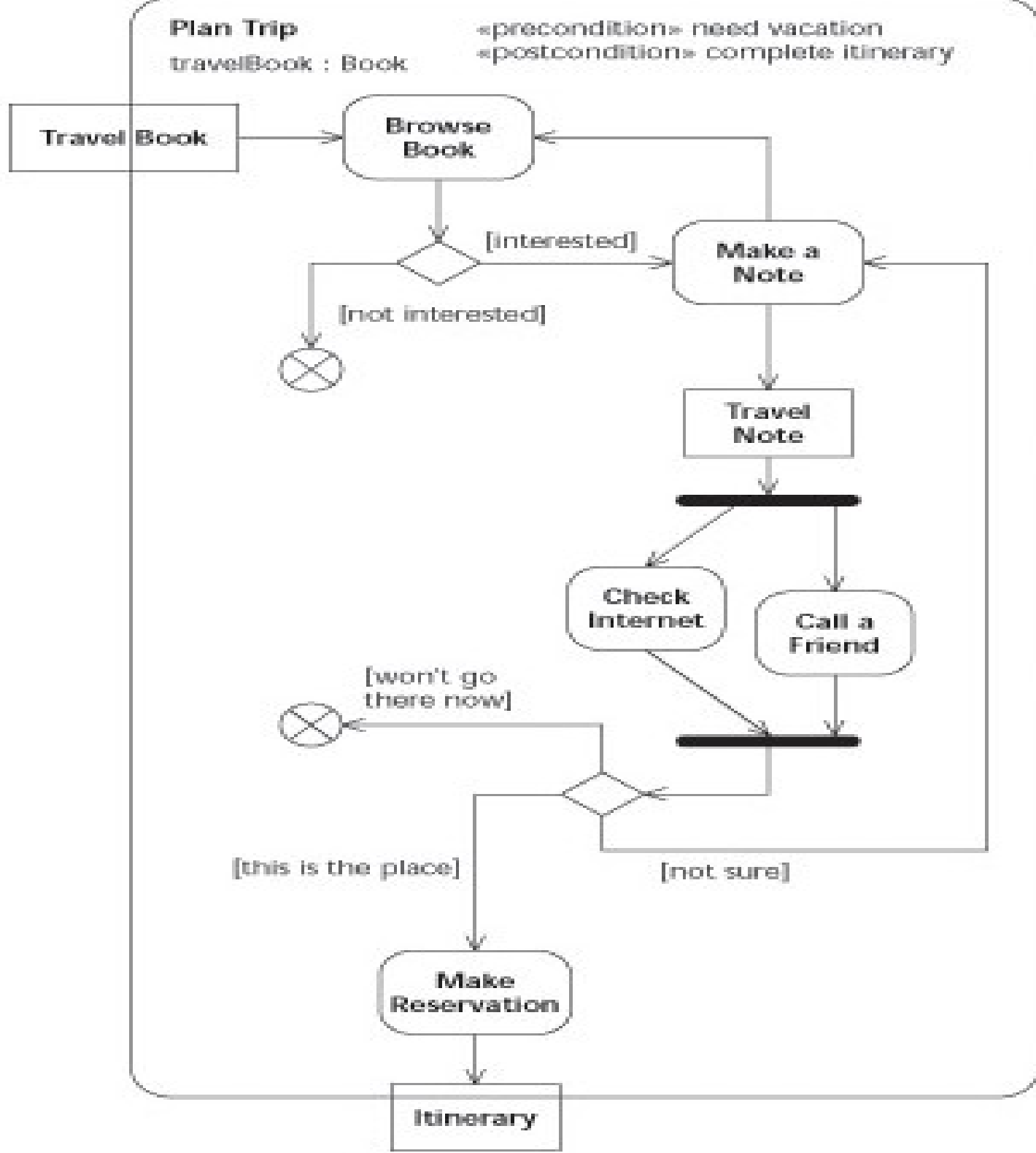
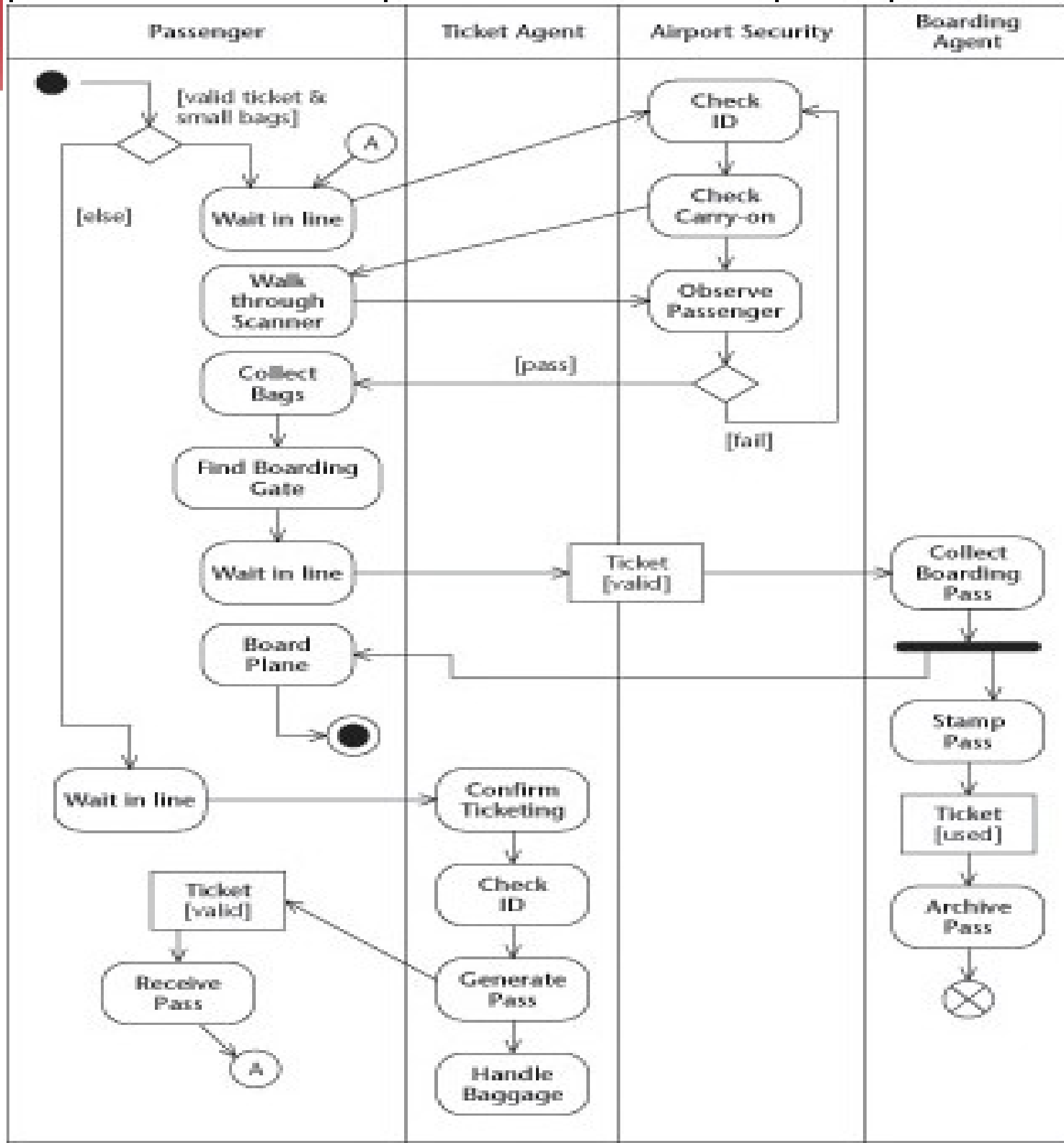


Figure 10.10.10. UML Activity Diagram for Plan Trip





# Latihan

- Buatlah activity diagram dari proses penerimaan mahasiswa baru
- Studi kasus: UKDW