Pengembangan Aplikasi Perangkat Lunak

OOAD

Activity Diagram

By: Augury El Rayeb (AER)

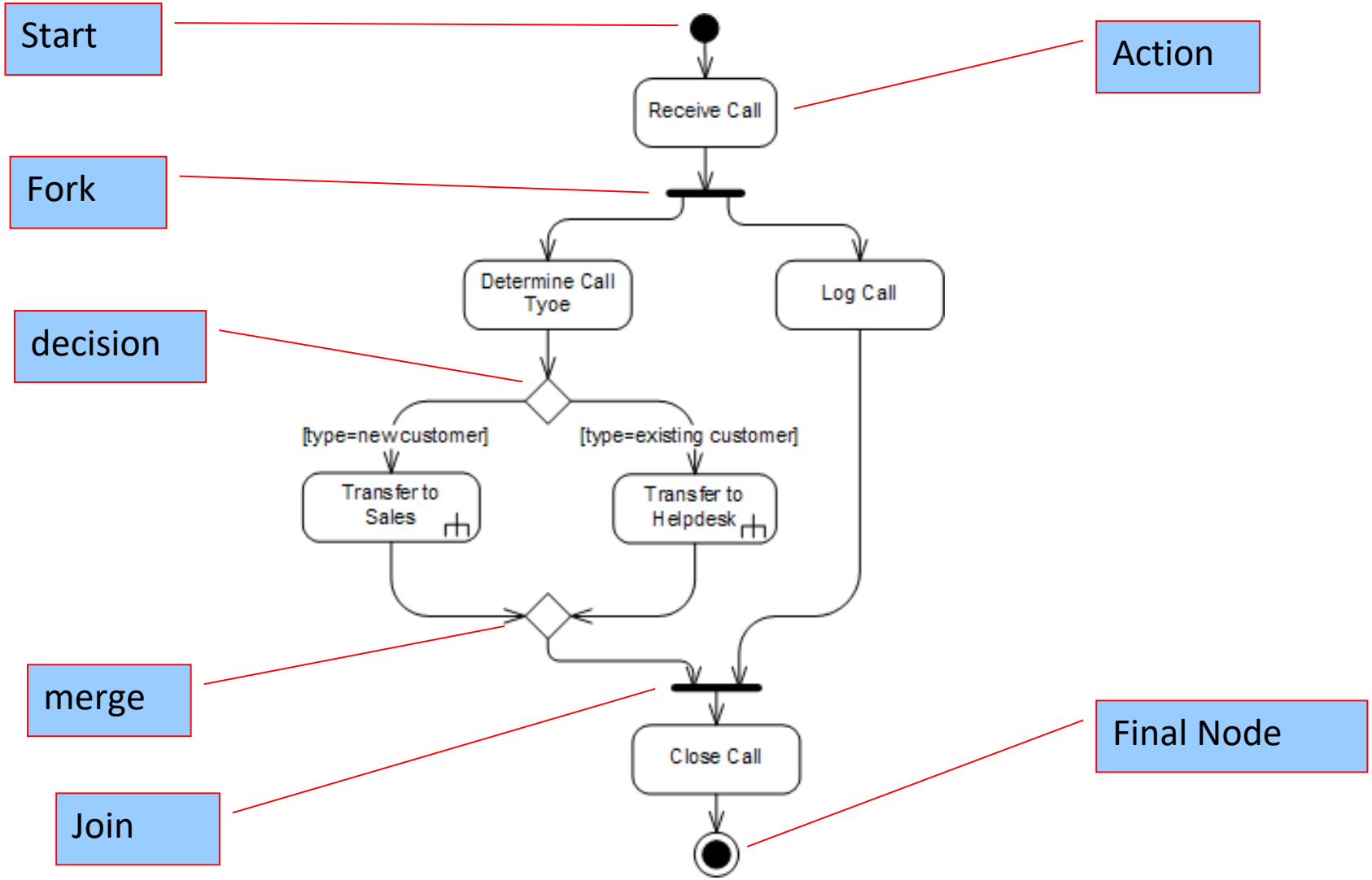
Tujuan Pertemuan

- Memahami pemodelan *procedural flow dari actions*, pemodelan *business-level function*, dan *system-level function*.
- Memahami kegunaan activity diagram
- Mampu membuat activity diagram

System Development Process

Phase	Actions	Outcome
Initiation	Raising a business need	Business documents
Requirements	Interviewing stakeholders, exploring the system environment	Organized documentation
Specification	Analyze the engineering aspect of the system, building system concepts	Formal specification
Design	Define architecture, components, data types, algorithms	Formal Specification
Implementation	Program, build, unit-testing, integrate, documentation	Testable system
Testing & Integration	Integrate all components, verification, validation, installation, guidance	Testing results, Working sys
Maintenance	Bug fixes, modifications, adaptation	System versions

Contoh Activity - Customer Service



Activity Diagram – What is it?

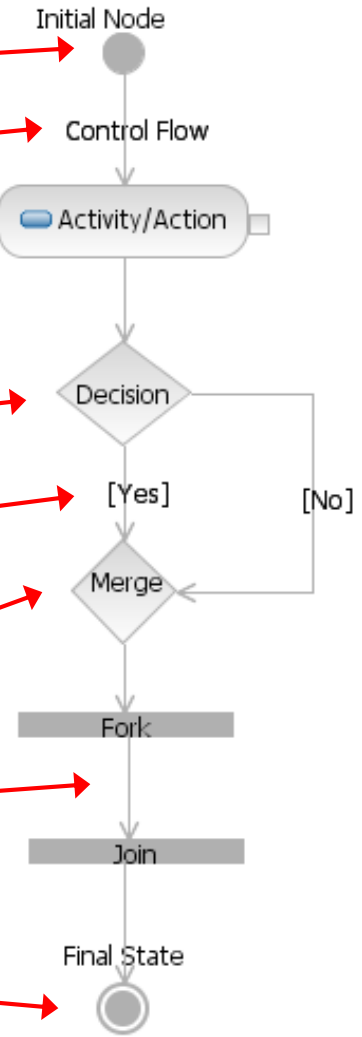
- Activity diagram digunakan untuk me-model-kan ***procedural flow dari actions*** (aksi) yang merupakan bagian dari suatu aktifitas yg besar.
- Activity diagram berfokus pada urutan aksi dari pelaksanaan dan kondisi yang memicu aksi tersebut.
- Activity diagram merupakan ***model specific (detailed level) dari use-case***.
- Activity diagram dapat digunakan secara independen (tanpa use-case) untuk me-model-kan suatu ***business-level function***
  (*contoh: proses/aktifitas pembelian tiket konser, proses pendaftaran, dsb*).
- Activity diagrams juga dapt digunakan untuk me-model-kan suatu ***system-level functions***.
  (*contoh: proses/aktifitas bagaimana data-mart reservasi tiket* how a ticket reservation data mart mempopulasi data warehouse sistem penjualan perusahaan.

How does the activity diagram relate to the use case template?

- Tiap komponen diagram activity merepresentasikan detil dari use-case, sbb:
  - Precondition – Initial state.
  - User step – Event.
  - System action – Activity.
  - Alternate flow – Decision.
  - Alternate flow returning to the basic flow – Merge.
  - System activities running in parallel – Fork.
  - Parallel activities joining the basic flow – Join.
  - Postcondition – Final node.

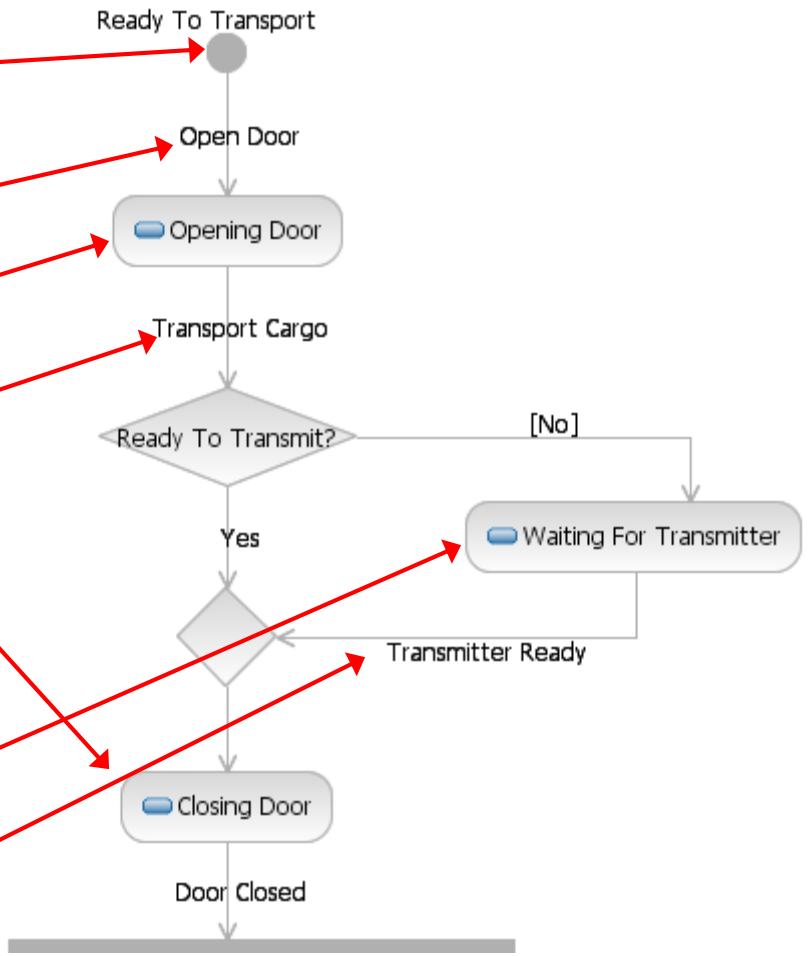
Pemetaan Activity – Use-case

- Precondition
- Actor input
- System Step
- Alternative or extension flow
- Basic Flow
- Returning alternate flow
- Parallel activities
- Postcondition



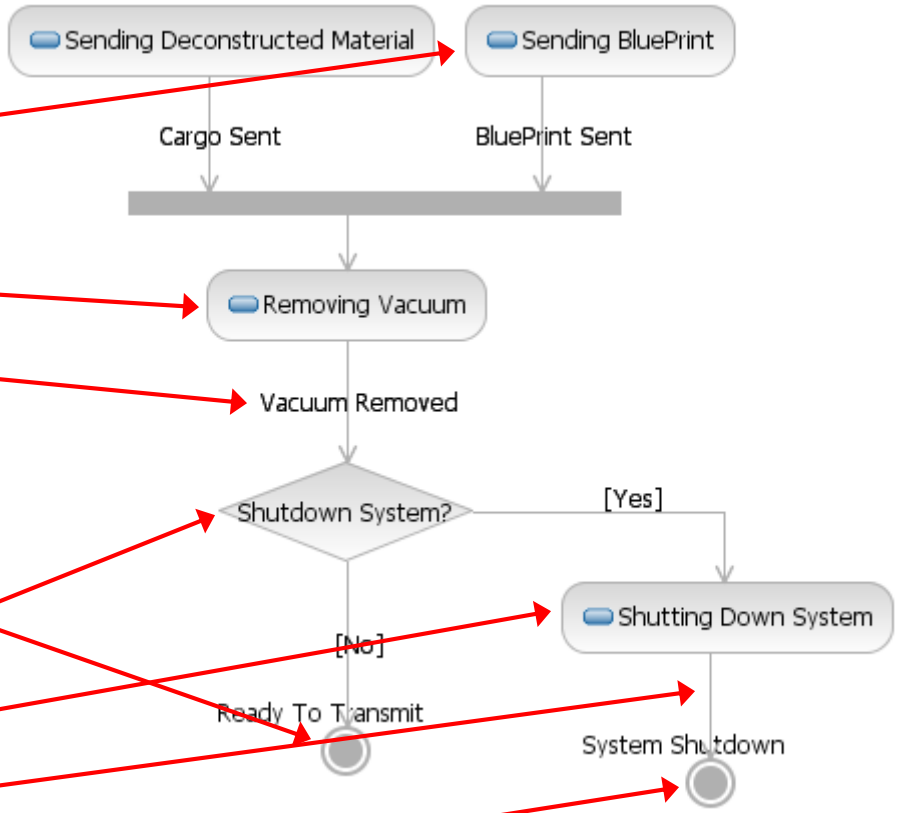
Contoh: Pemetaan Activity – Use-case

- Precondition
  - The system is ready to transport cargo.
- Basic Flow
  1. The system receives a command to open the door to the Dematerializer.
  2. The system opens the door to the Dematerializer.
  3. The system receives a command to transport the cargo.
  4. The Transmitter is ready to transmit and the system closes the door to the Dematerializer.
- Alternate Flow
  1. At step 4, the Transmitter is not ready to transmit, and the system waits for the transmitter to be ready to transmit.
  2. The Transmitter is ready to transmit and the use case continues from step 4.



Contoh: Pemetaan Activity – Use-case (lanjutan)

- Basic Flow (continued)
  6. The system sends the deconstructed cargo to the Transmitter and sends the cargo blueprint to the Blueprint Manager.
  7. The system removes the vacuum from the Dematerializer.
  8. The vacuum is removed and the use case ends.
- Postcondition
  - The system is ready to transmit cargo.
- Extension Flow
  1. At step 8, the system is set to shutdown when the cargo is transmitted, and the systems performs a shutdown.
  2. The system shutsdown and the use case ends.
- Postcondition
  - The system is shutdown



Model Konseptual Activity Diagram

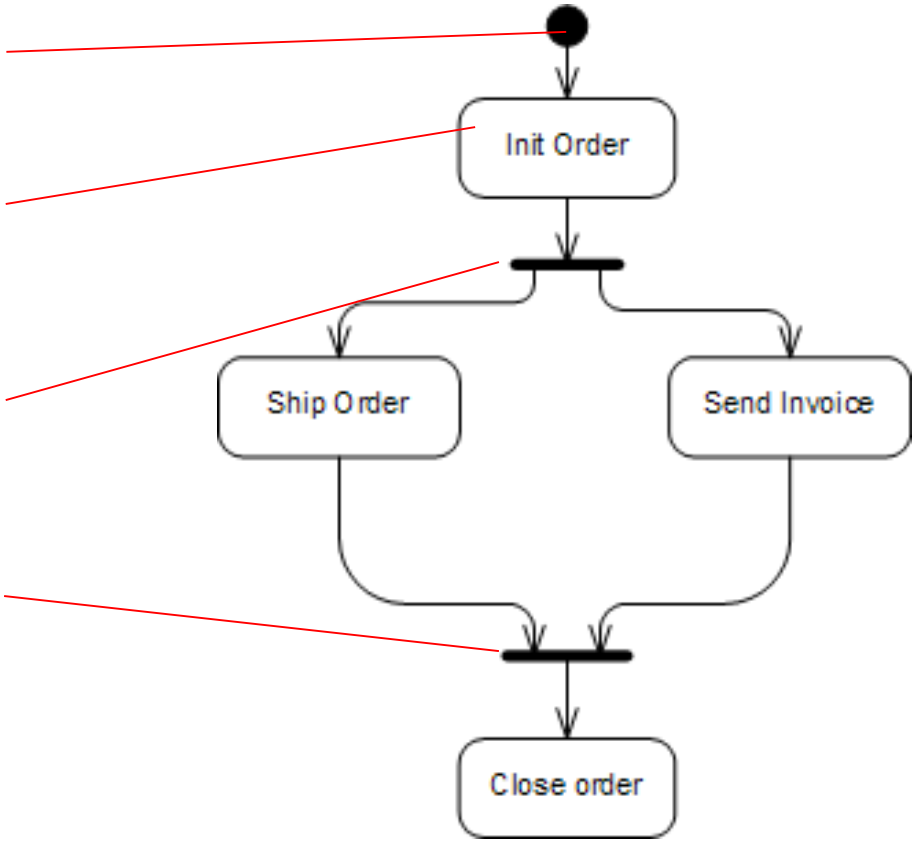
- Model konseptual activity diagram berbasiskan pada **token** (pemicu)**.**

Node inisial membuat single token (pemicu tunggal)

Suatu action membutuhkan token utk mulai mengerjakan, dan memproduksi token berikutnya

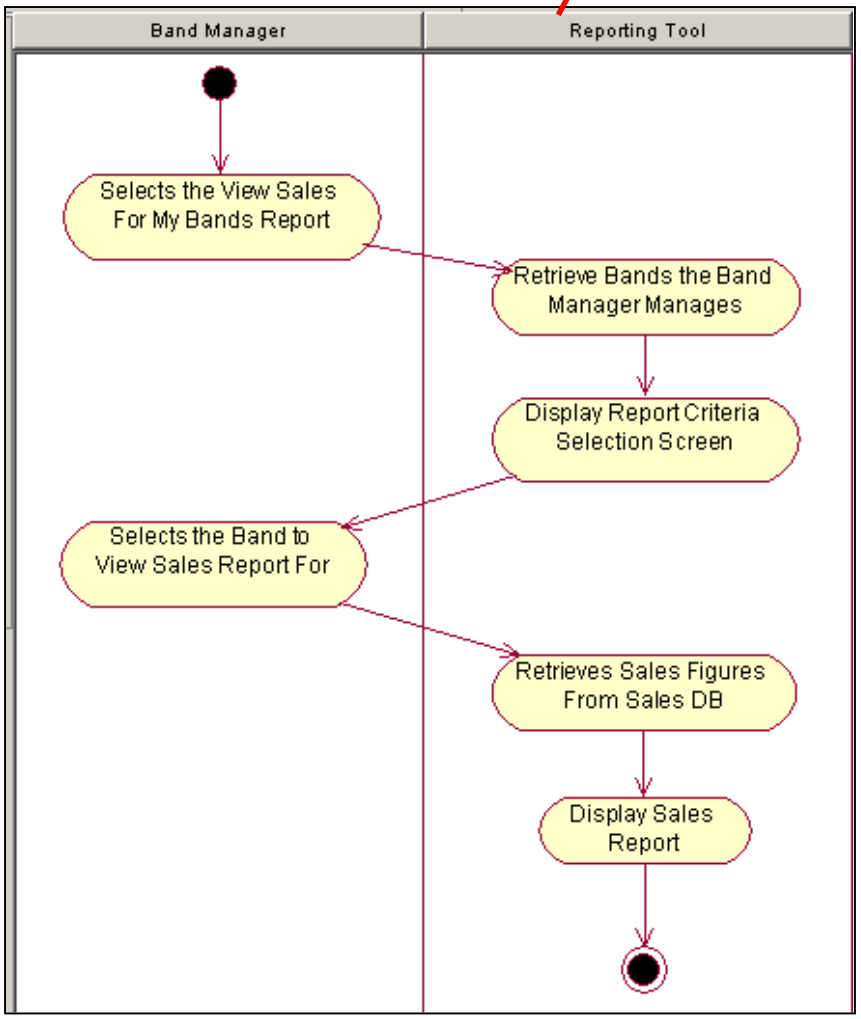
Setiap node fork menghasilakn token, sesuai jumlah jalur.

Tiap join “collects” semua tokens yang masuk, kemudian memproduksi token selanjutnya.

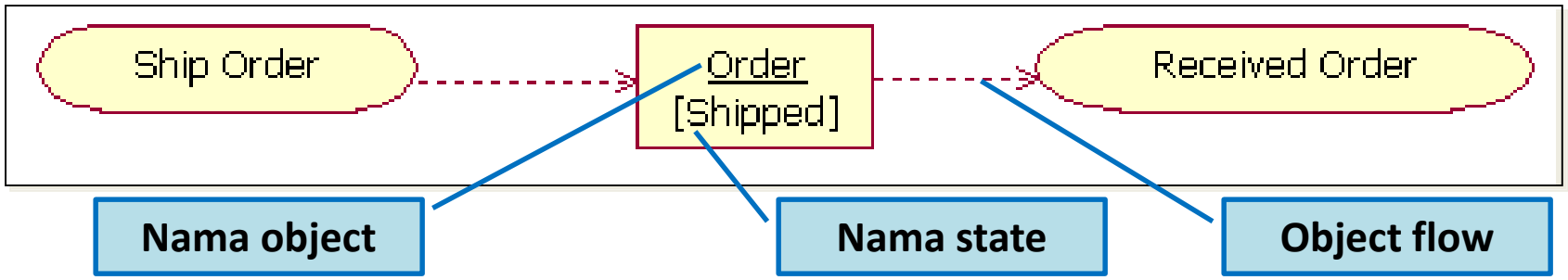


Activity Diagram - Swimlanes

- Dalam me-model-kan *activity's procedural flow* sering diperlukan representasi dari kendali eksekusi pada tiap-tiap object (persons, organizations, or other responsible entities).
- **Swimlanes** merupakan garis batas yang menyatakan wilayah dari object-object dalam eksekusi suatu aksi.

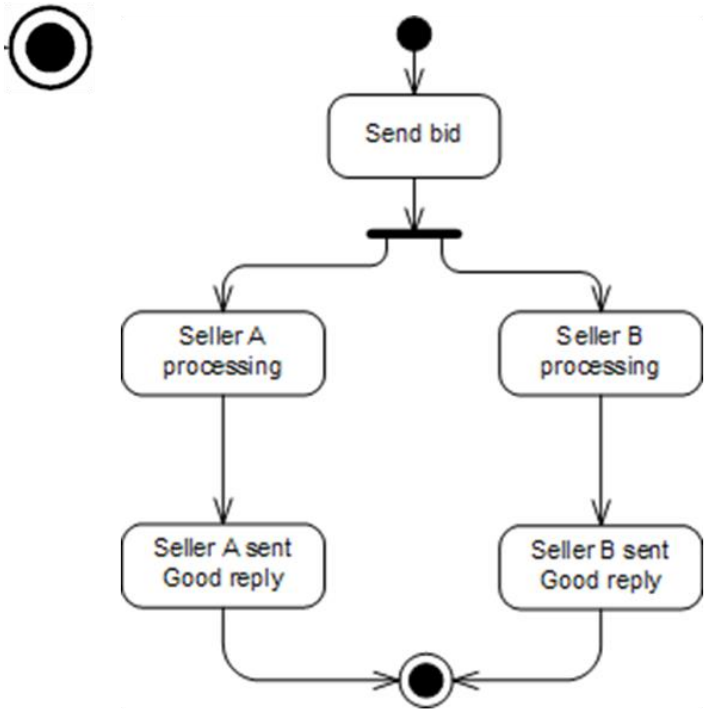
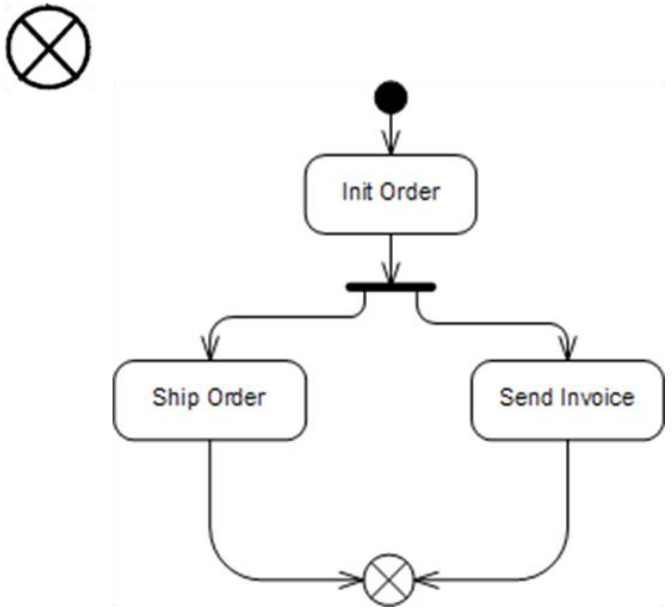


Activity Diagram – Object on State



- Objects digunakan untuk **identifikasi analisis class** potensial
- **Object pada suatu state (kondisi)**, dalam activity diagram digambarkan dengan symbol kotak.
- Penulisan ***nama object*** menggunakan ***underline***
- Teks dalam kurung kotak (**[…..]**) merupakan ***nama state (keadaan)*** dari object pada saat itu.
- Object flow digambarkan dengan dashed line arrow.

Final Nodes

the activity is terminated when the first token arrives	the activity is terminated when all tokens in the graph are destroyed
	

Activity Diagram - Contoh Lengkap

- Contoh activity diagram lengkap

Take home MidTest

- Buat use-case yang menunjukkan customer melakukan transaksi pada ATM (ambil uang dan setor uang).
- Buat pemodelan use-case ATM tersebut dengan diagram Activity.
- There are no ‘right’ answers.
- There is no industry standard for modeling use cases.

Thank's

- See Ya Next Week