

# MATERI 4 - *Activity Diagram* APSI - 2 (Analisis & Perancangan Sistem Informasi -2)

---

ANNISA PARAMITHA F., S.KOM., M.KOM  
PRODI SISTEM INFORMASI & MANAJEMEN INFORMATIKA  
UNIKOM

## Definisi *Activity Diagram*

*Activity Diagram* adalah bagian penting dari UML yang menggambarkan aspek dinamis dari sistem. Logika prosedural, proses bisnis, dan aliran kerja suatu bisnis dapat dengan mudah dideskripsikan dalam *activity diagram*.

Tujuan dari *activity diagram* adalah untuk menangkap tingkah laku dinamis dari sistem dengan cara menunjukkan aliran pesan dari satu aktivitas ke aktivitas lain.

Secara umum tujuan dari *activity diagram* bisa digambarkan sebagai berikut :

- Menggambarkan aliran aktivitas dari sistem.
- Menggambarkan urutan aktivitas dari satu aktivitas ke lainnya.
- Menggambarkan paralelisme, percabangan dan aliran konkuren dari sistem.

## Definisi *Activity Diagram*

Secara umum *Activity Diagram* digunakan untuk menggambarkan diagram alir yang terdiri dari banyak aktivitas dalam sistem dengan beberapa fungsi tambahan seperti : percabangan, aliran parallel, swim lane dsb

Penggunaan *activity diagram* mencakup hal - hal berikut :

- Pemodelan aliran kerja yang menggunakan aktivitas
- Pemodelan kebutuhan bisnis
- Pemahaman tingkat tinggi dari fungsionalitas sistem.

# Simbol *Activity Diagram* (Aktivitas)



Aktivitas

Aktivitas adalah fungsi yang dilakukan oleh sistem.

Aktivitas adalah perilaku parameter yang direpresentasikan sebagai aliran tindakan yang terkoordinasi.

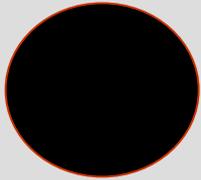
Aktivitas dapat berupa :

- *Action*
- *Object*
- *Control*

Kegiatan (*Activity*) mungkin berisi tindakan (*Actions*) dari berbagai jenis:

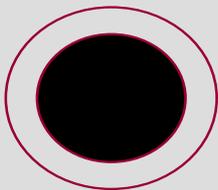
- Kemunculan fungsi primitif, seperti fungsi aritmatika.
- Perilaku (*Behavior*).
- Manipulasi objek, seperti membaca atau menulis atribut atau asosiasi.

# Simbol *Activity Diagram*



*Start State/Activity Initial Node*

Menunjukkan dimulainya suatu *workflow* pada sebuah *activity diagram*. Hanya ada satu *Initial Node* dalam satu *workflow*



*End State/Activity Final Node*

End state menggambarkan akhir atau terminal dari pada sebuah *activity diagram*.

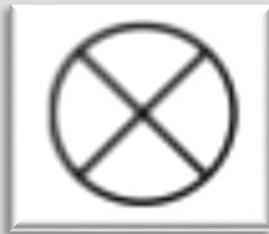
*Activity Final Node* adalah simpul akhir kontrol yang menghentikan semua aliran aktivitas dalam *workflow*. Dalam satu *workflow* dapat memiliki lebih dari satu *Final Node*

# Simbol *Activity Diagram*



*State Transition*

*Transitions* ketika sebuah aktivitas atau state selesai, maka *flow control* berganti ke aktivitas atau *state* berikutnya.



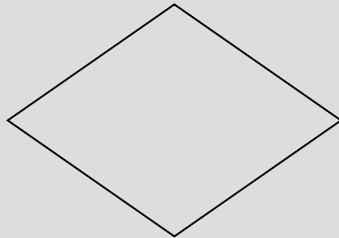
*Flow Final Node*

*Flow Final Node* menunjukkan akhir sebuah flow tertentu, tanpa menghentikan seluruh *activity*.

Semua node yg mengarah ke *Final Node* akan hancur/ berhenti, tapi tidak akan berpengaruh ke aktivitas lain di dalam *workflow*.

# Simbol *Activity Diagram*

## Decision



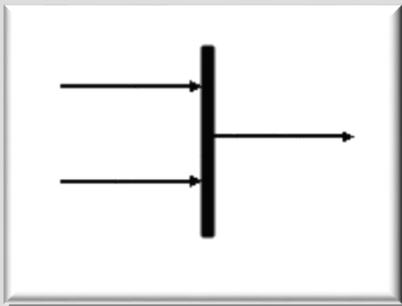
*Decision* adalah suatu titik/point pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi.

Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

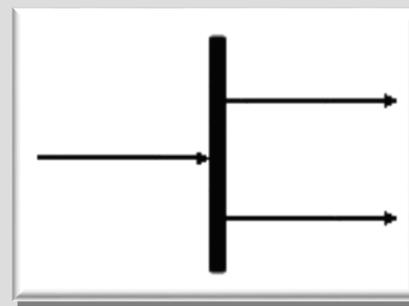
# Simbol *Activity Diagram*

**Fork Node** adalah node kontrol yang memiliki satu aliran masuk dan membagi menjadi beberapa aliran keluar, atau sebaliknya.

Menggabungkan beberapa aliran aktivitas menjadi satu aliran aktivitas atau sebaliknya

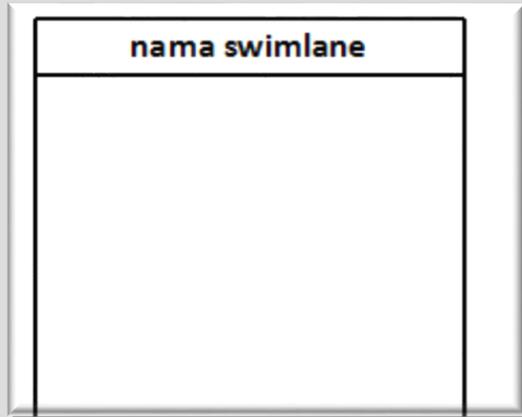


**Join** (Penggabungan)

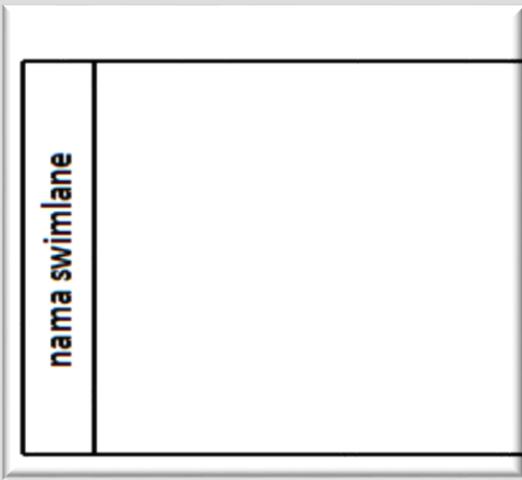


**Fork** (Percabangan)

# Simbol *Activity Diagram*

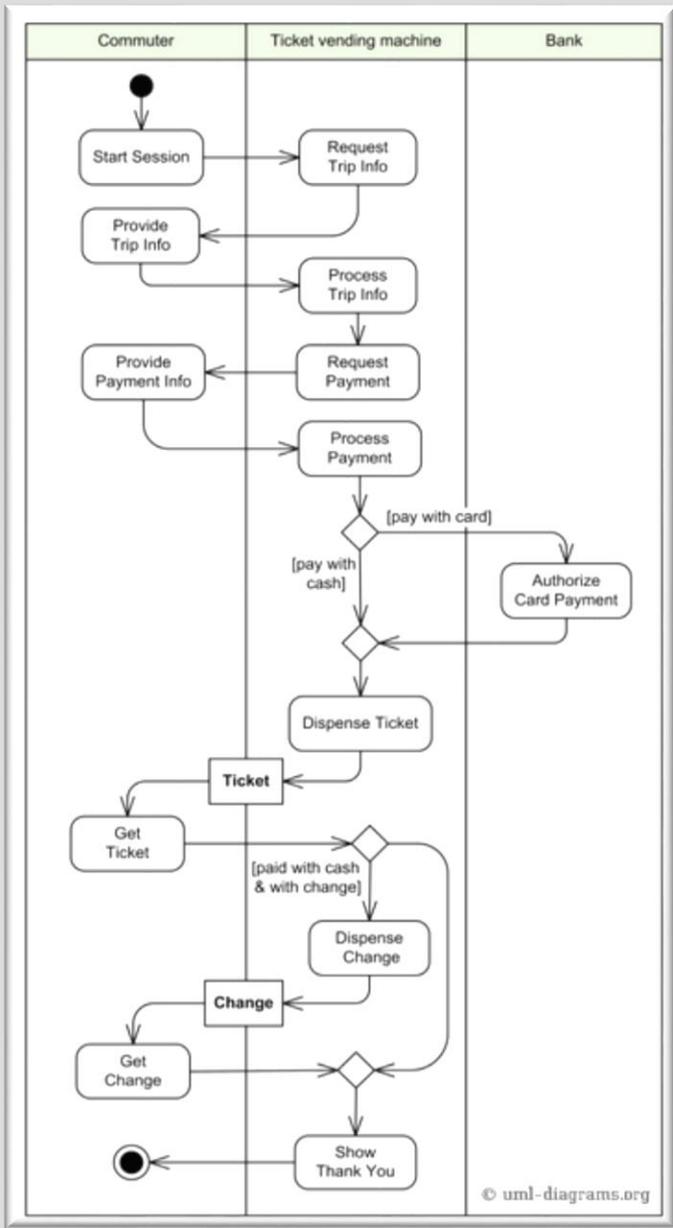


atau



Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

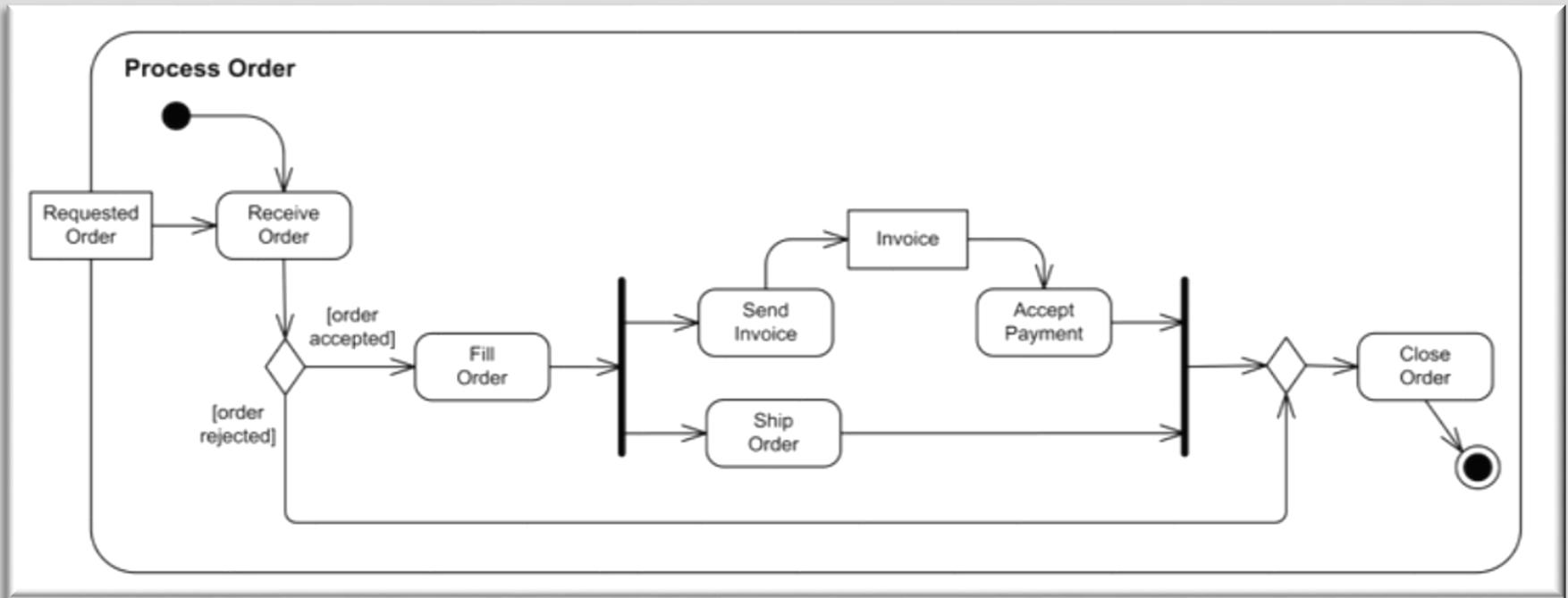
Sebuah cara mengelompokan aktivitas berdasarkan aktor



Ini adalah contoh *Activity Diagram* UML yang menjelaskan Use Case Pembelian Tiket .

Kegiatan dimulai oleh *Commuter Actor* yang perlu membeli tiket. Mesin penjual tiket akan meminta informasi perjalanan dari *Commuter*. Informasi ini akan mencakup jumlah dan jenis tiket, misalnya apakah itu tiket bulanan, tiket sekali jalan atau tiket pulang-pergi, nomor rute, nomor tujuan atau zona, dll.

Berdasarkan mesin penjual tiket info perjalanan yang disediakan akan menghitung total pembayaran dan meminta opsi pembayaran. Opsi-opsi itu termasuk pembayaran dengan uang tunai, atau dengan kartu kredit atau kartu debit. Jika pembayaran dengan kartu dipilih oleh *Commuter*, aktor lain, yaitu Bank akan berpartisipasi dalam aktivitas dengan mengotorisasi pembayaran.



*Request Order* adalah parameter input dari aktivitas. Setelah pesanan diterima dan semua informasi yang diperlukan diisi, pembayaran diterima dan pesanan dikirim. Perhatikan, bahwa aliran bisnis ini memungkinkan pengiriman pesanan sebelum faktur dikirim atau pembayaran dikonfirmasi.

# REFERENSI

- Munawar, Analisis Perancangan Sistem Berorientasi Objek dengan UML (*Unified Modelling Language*), Bandung, 2018, Penerbit Informatika
- <https://www.uml-diagrams.org/activity-diagrams.html>
- <https://www.uml-diagrams.org/activity-diagrams-controls.html>