

The image features two large, thick black L-shaped brackets. One is positioned in the top-left corner, and the other is in the bottom-right corner. They are oriented towards each other, framing the central text.

ACTIVITY DIAGRAM

Pendahuluan

- Activity Diagram (Activity Diagram) diperlukan untuk menggambarkan **proses bisnis** dan **urutan aktivitas** dalam sebuah proses
- Activity Diagram juga dipakai pada *business modeling* untuk memperlihatkan **urutan aktivitas** proses bisnis
- Struktur diagram ini mirip flowchart atau Data Flow Diagram pada perancangan terstruktur
- Activity Diagram sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan
- Activity diagram dibuat berdasarkan **sebuah** atau **beberapa use case** pada use case diagram

- Activity Diagram berhubungan dengan diagram Statechart. Diagram Statechart fokus pada *obyek dalam suatu proses* (atau proses menjadi suatu obyek), sedangkan Activity Diagram fokus pada *aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal*.
- Dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain
- Yang perlu diperhatikan disini adalah **bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor**

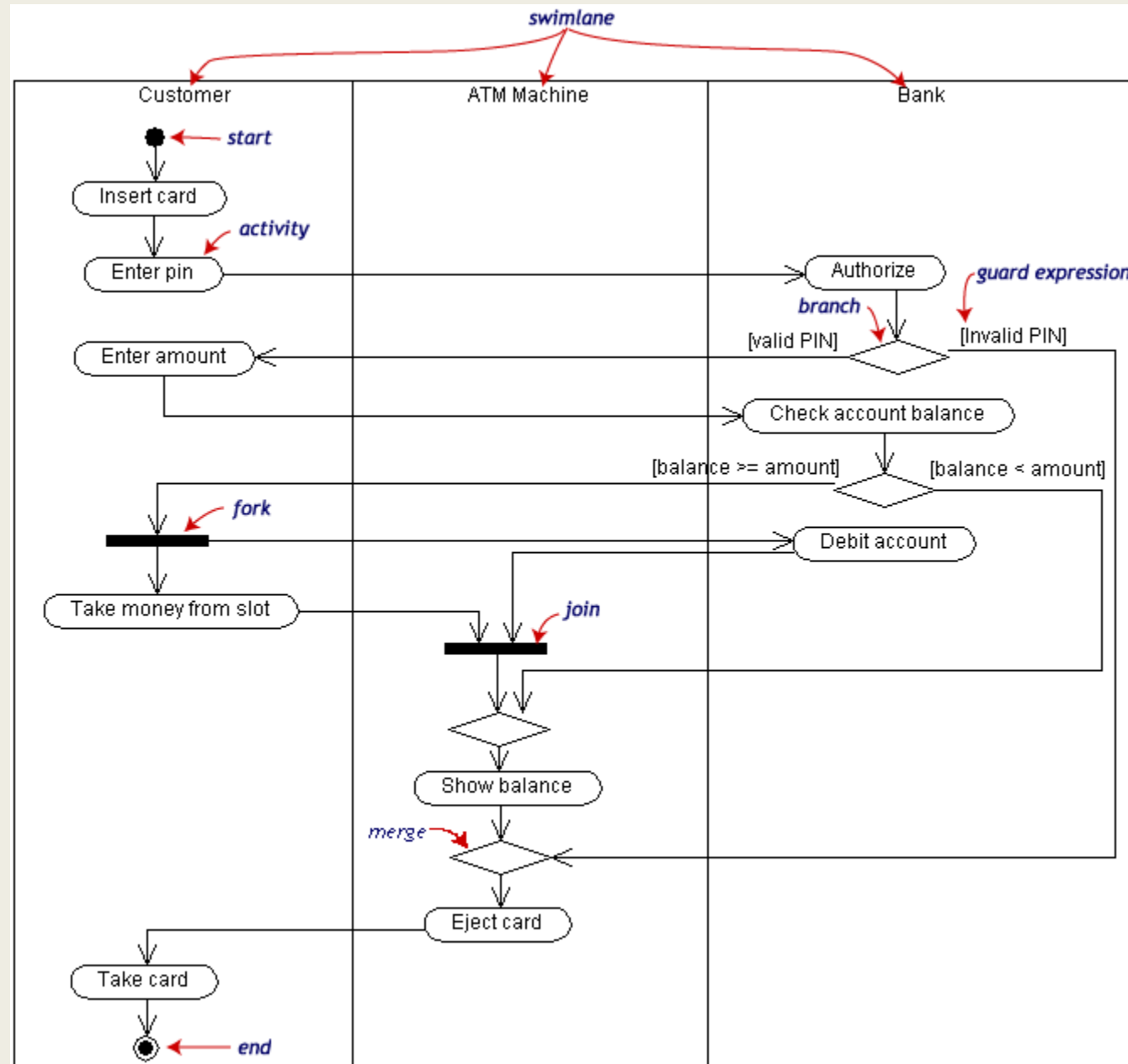
- *Activity Diagram* merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*).
- Oleh karena itu *Activity Diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.
- Sebuah **aktivitas dapat direalisasikan oleh satu use case atau lebih**. **Aktivitas** menggambarkan **proses yang berjalan**, sementara **use case** menggambarkan bagaimana **aktor** menggunakan sistem untuk melakukan aktivitas.

Contoh Activity Diagram




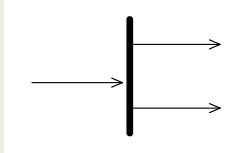
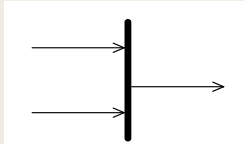

- “Pengambilan uang dari bank melalui ATM”
- Ada tiga aktifitas kelas (orang, dan lainnya) yang terkait, yaitu : Customer, ATM, and Bank.
- Proses berawal dari lingkaran start hitam pada bagian atas dan berakhir di pusat lingkaran stop hitam/putih pada bagian bawah. Aktivitas digambarkan dalam bentuk kotak persegi.
- Lihat gambar agar lebih jelas :

CONTOH ACTIVITY DIAGRAM

Penarikan Uang dari Account Bank Melalui ATM

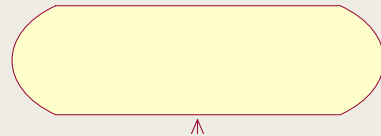


Simbol Activity Diagram

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)

Activity

- Activity menggambarkan sebuah pekerjaan/tugas dalam workflow.
- Pada UML, activity digambarkan dengan simbol belah ketupat='lozenge' (horizontal top and bottom with convex sides).



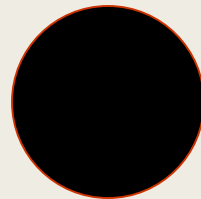
Activity



State

Start State

- Start state dengan tegas menunjukkan dimulainya suatu workflow pada sebuah activity diagram
- Hanya ada satu start state dalam sebuah workflow
- Pada UML, start state digambarkan dengan simbol lingkaran yang solid

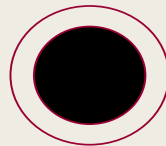


Start State

End State

- End state menggambarkan akhir atau terminal dari pada sebuah activity diagram
- Bisa terdapat lebih dari satu end state pada sebuah activity diagram
- Pada UML, end state digambarkan dengan simbol sebuah bull's eye (mata sapi)

End State



State Transitions

- State transition menunjukkan kegiatan apa berikutnya setelah suatu kegiatan sebelumnya
- Pada UML, state transition digambarkan oleh sebuah *solid line* dengan panah

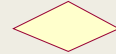


State Transition

Decisions

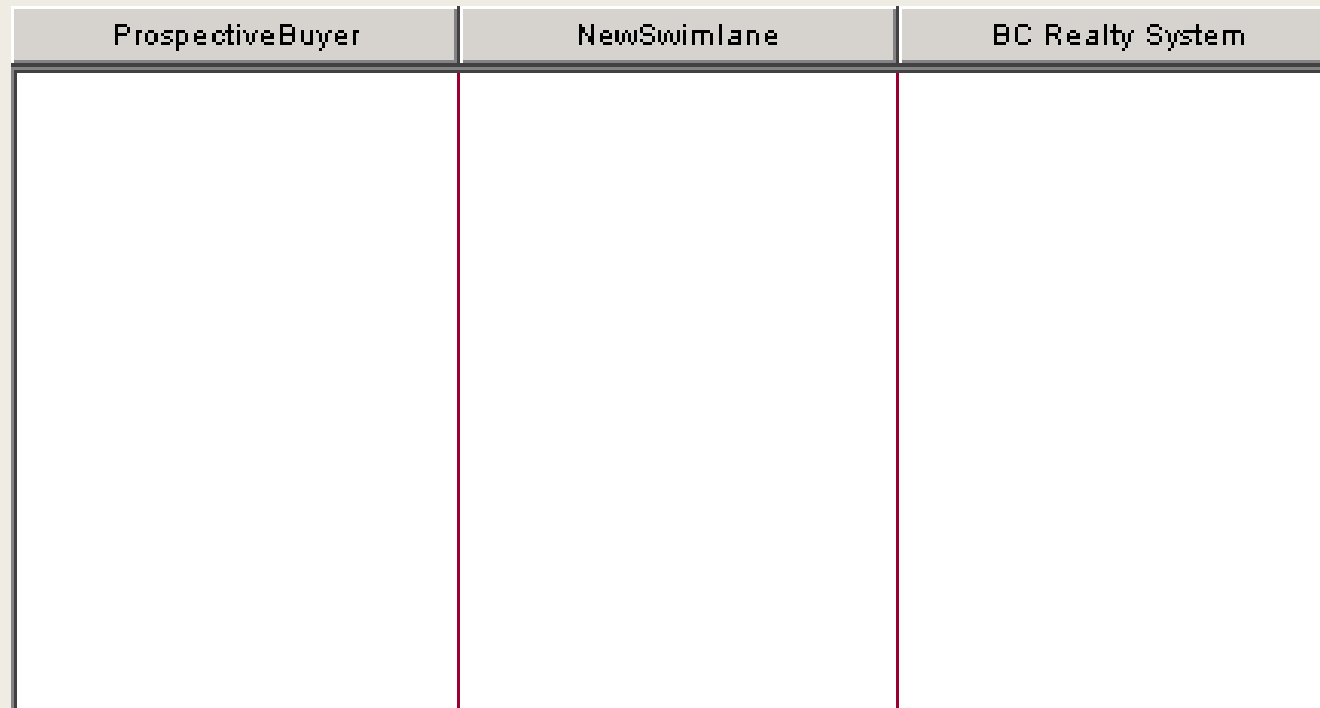
- Decision adalah suatu titik/point pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi
- Pada UML, decision digambarkan dengan sebuah simbol diamond

Decision



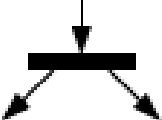

Swimlanes

- *Object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

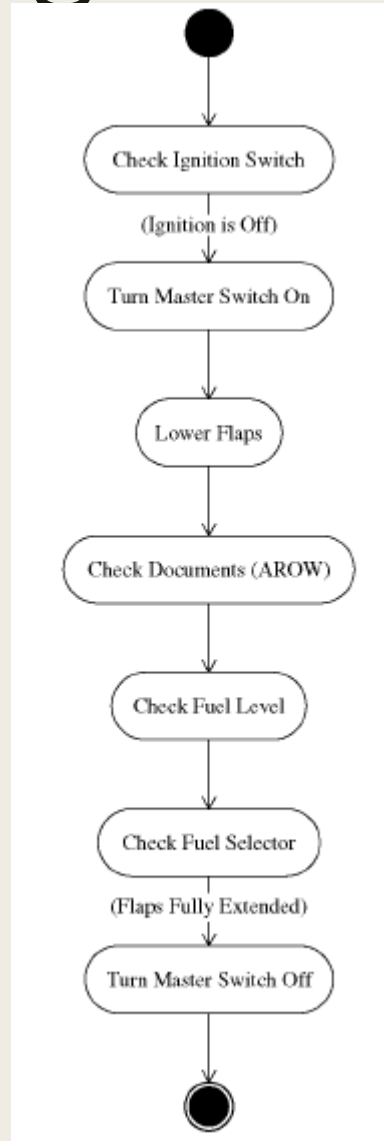


Fork dan Join

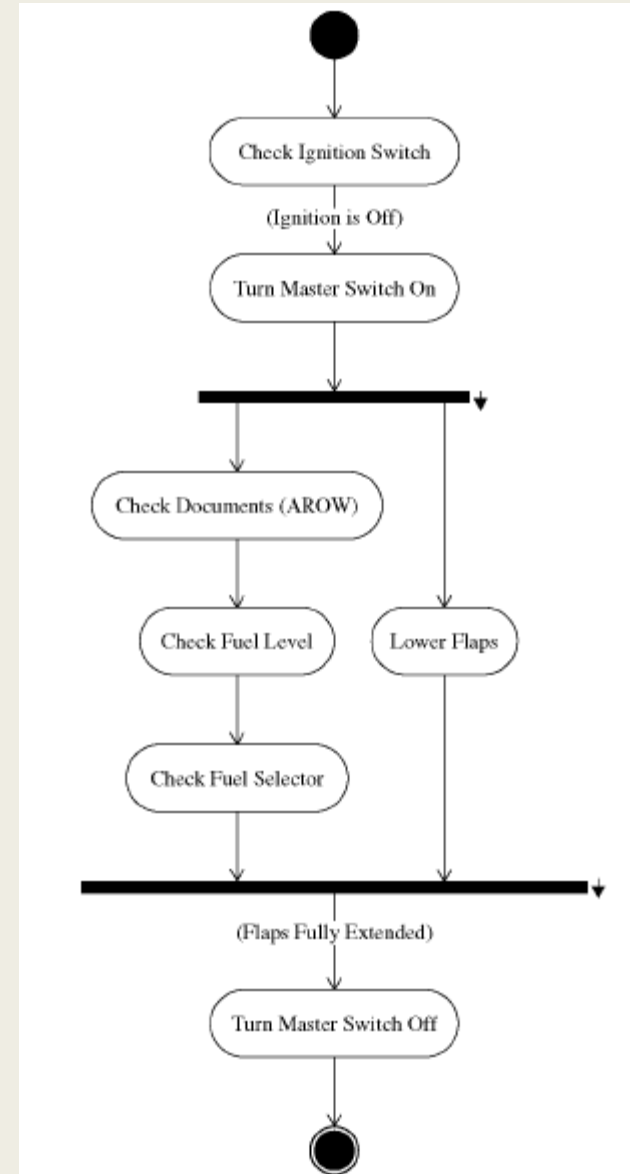
- Diagram Activity dapat dibagi menjadi beberapa jalur kelompok yang menunjukkan obyek yang mana yang bertanggung jawab untuk suatu aktifitas.
- Peralihan tunggal (*single transition*) timbul dari setiap adanya *activity* (aktifitas), yang saling menghubungkan pada aktifitas berikutnya.
- Sebuah *transition* (transisi) dapat membuat cabang ke dua atau lebih percabangan *exclusive transition* (transisi eksklusif).
- Label *Guard Expression* (ada di dalam []) yang menerangkan output (keluaran) dari percabangan.
- Percabangan akan menghasilkan bentuk menyerupai bentuk intan. *Transition* bisa bercabang menjadi beberapa aktifitas paralel yang disebut **Fork**. *Fork* beserta *join* (gabungan dari hasil output *fork*) dalam diagram berbentuk *solid bar* (batang penuh).

Simbol	Deskripsi
	<i>fork</i> , digunakan utk menunjukkan kegiatan yg dilakukan secara paralel
	<i>join</i> , digunakan utk menunjukkan kegiatan yg digabungkan

Tanpa Percabangan



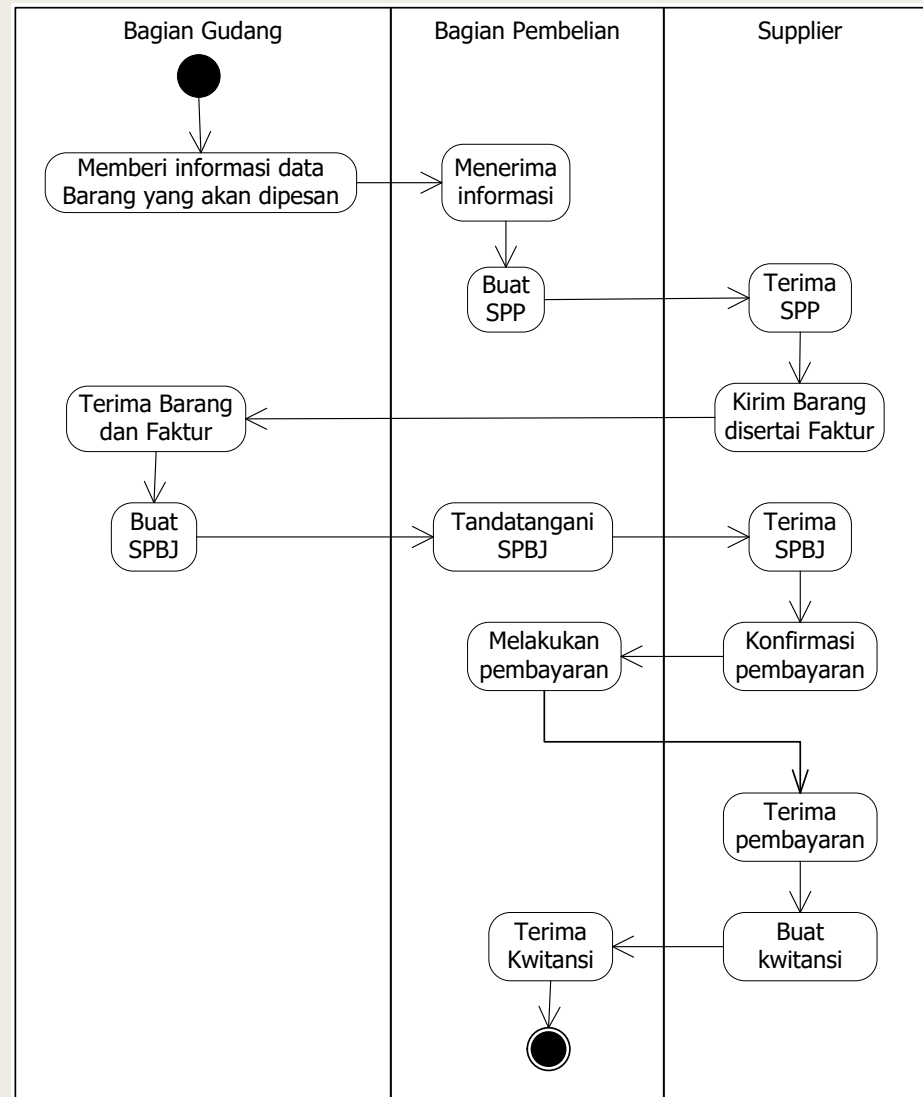
Dengan Percabangan



Petunjuk Membuat Diagram Aktivitas

1. Mulailah dengan **node awal** untuk titik awal
2. Tambahkan partisi jika relevan untuk **analisis** yang dibuat
3. Tambahkan **aksi** untuk setiap langkah utama dari use case
4. Tambahkan **alur dari setiap aksi ke aksi** lain, keputusan atau node akhir. Setiap aksi hanya mendapat **satu alur masuk** dan **satu alur keluar** menuju ke forks, joins, decisions, dan merges
5. Tambahkan **decisions** jika alur dipecah menjadi Activity Diagrami beberapa pilihan. Jangan lupa untuk menggabungkan kembali dengan merge
6. Tambahkan **forks dan joins** jika aktivitas akan dilakukan secara **paralel**
7. Akhiri proses dengan notasi untuk akhir aktivitas

Contoh Activity Diagram



Contoh Diagram Aktivitas Sistem Order

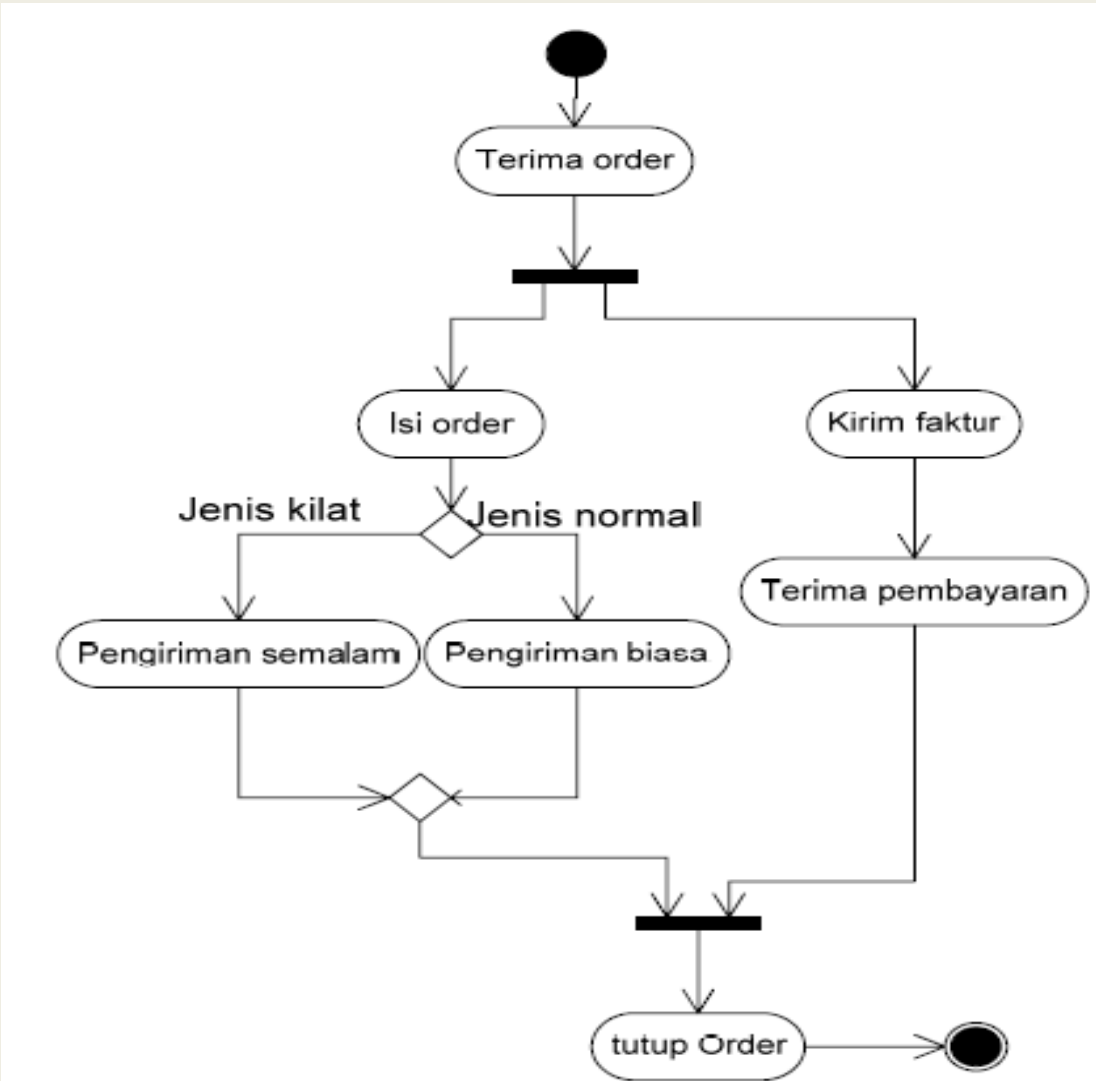
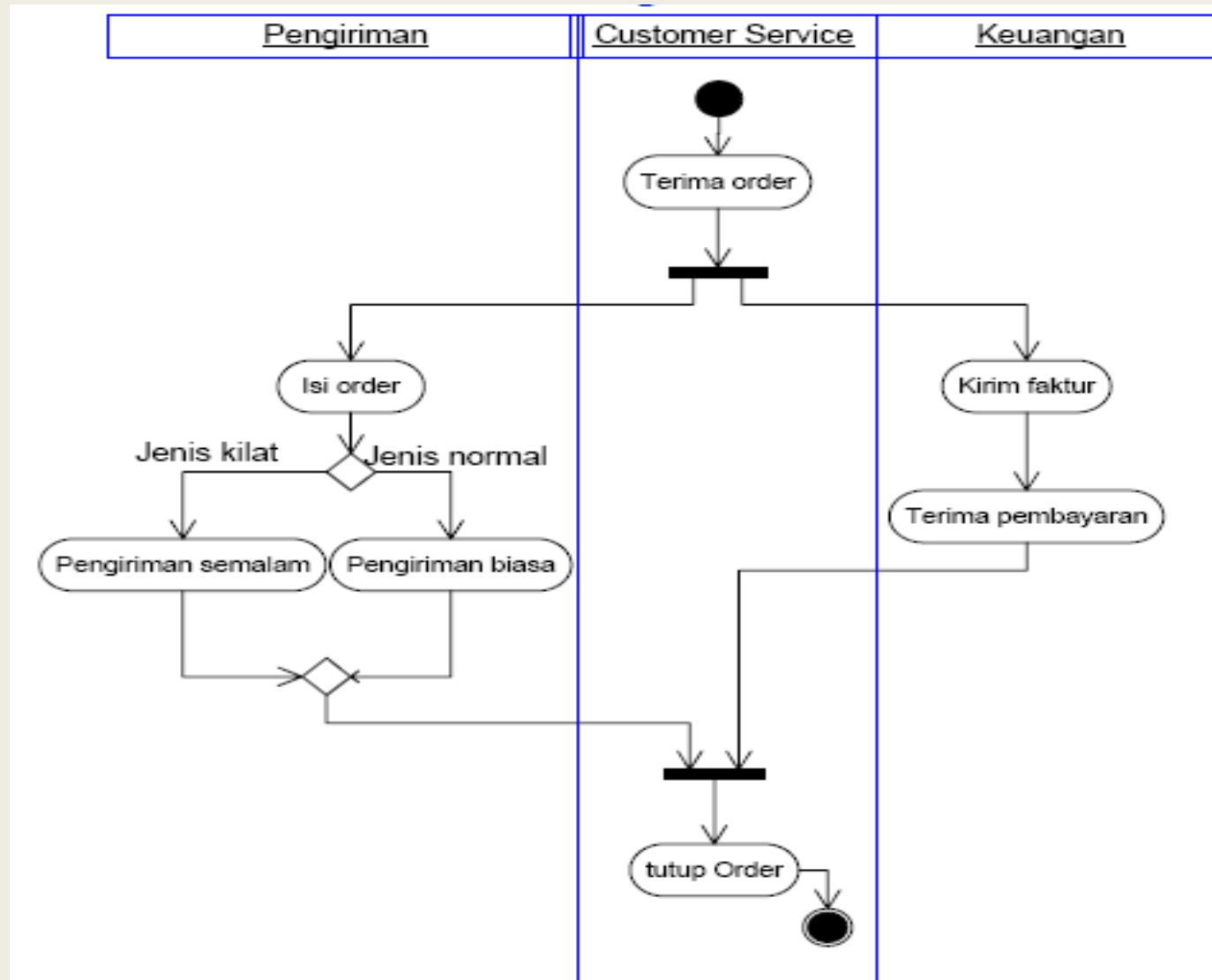
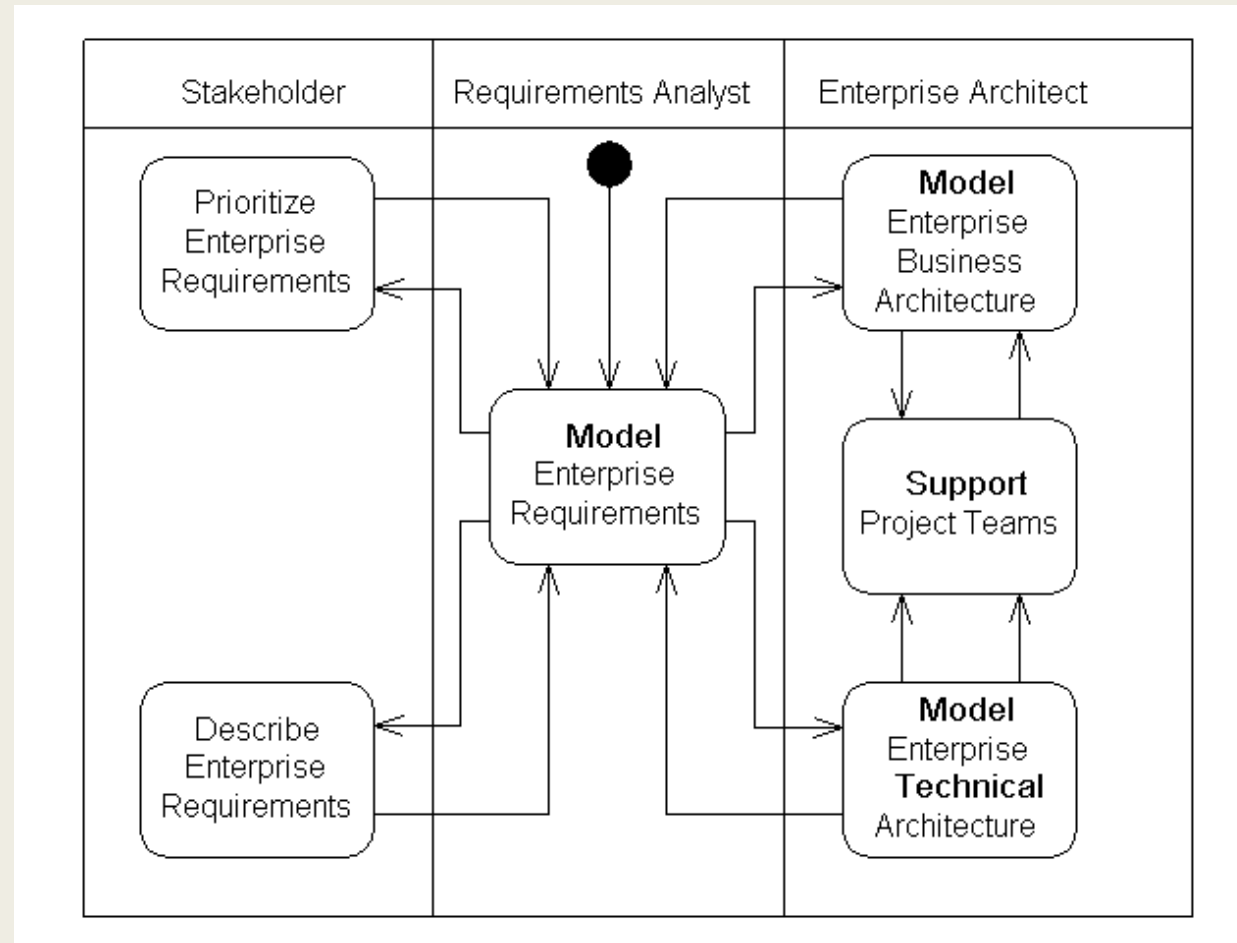


Diagram aktivitas sistem order dengan *Swimlane*



Activity diagram



Tips

- Identify the **use cases** that are **most critical**, and **create activity diagrams** for those use cases that require some exploration.
- If you are not sure how many activity diagrams to create, then try creating an activity diagram for each of the primary functions of your most important use cases.

Referensi

- Catur Iswahyudi + Edhy Sutanta, UML
- <http://www.agilemodeling.com/>
- <http://www.visual-parActivityDiagramigm.com/VPGallery/diagrams/index.html>