

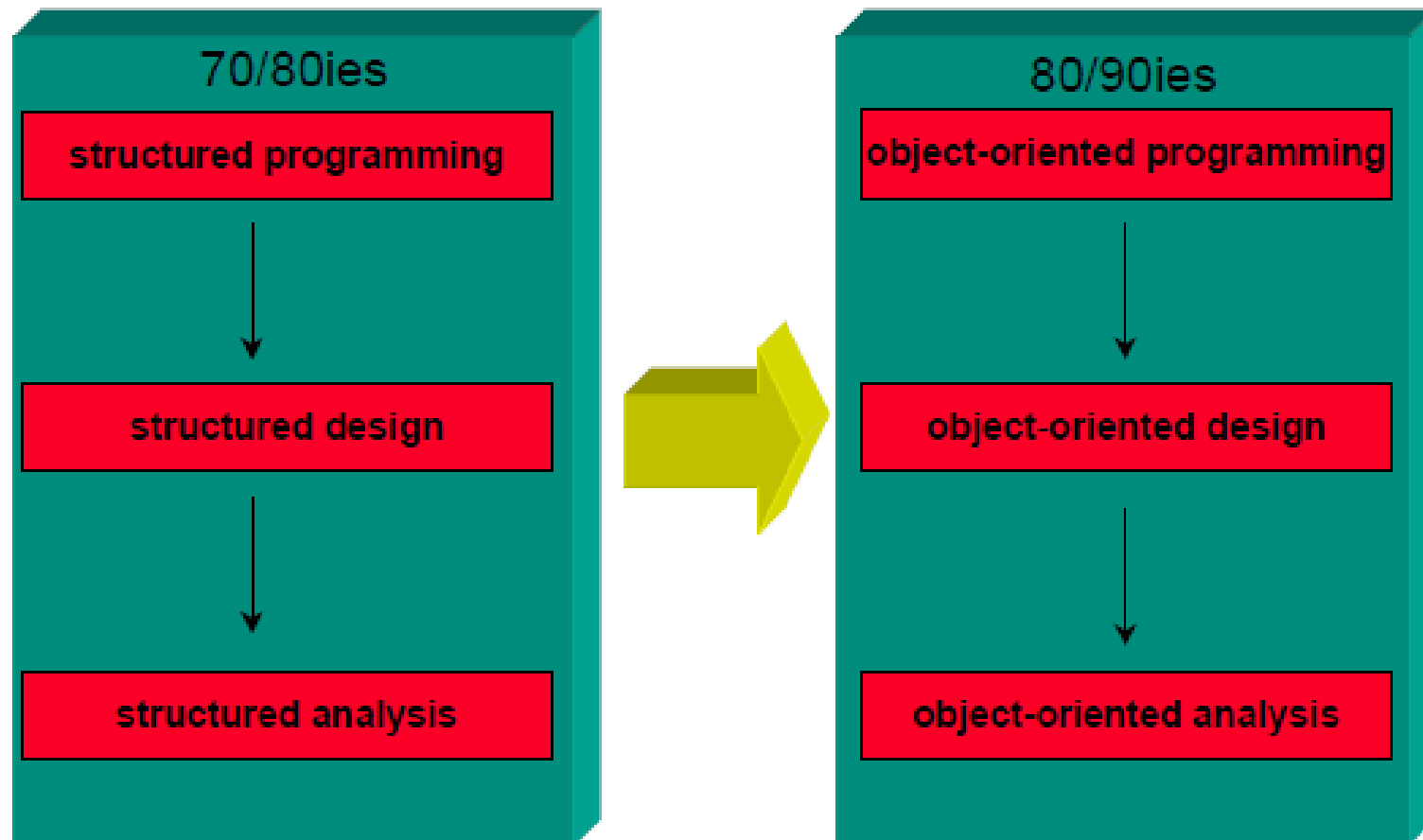


Perancangan Berorientasi Objek -Desain dan Analisis Sistem -

NISA'UL HAFIDHOH, M.T.

PRODI DIII – TEKNOLOGI INFORMASI

Evolusi Metode Pengembangan Sistem



Sejarah OOAD hingga UML

1970

First object-oriented languages (Simula-67, Smalltalk).

1980

More than 50 different OOAD languages cause the users trouble to find complete and appropriate tools.

1992

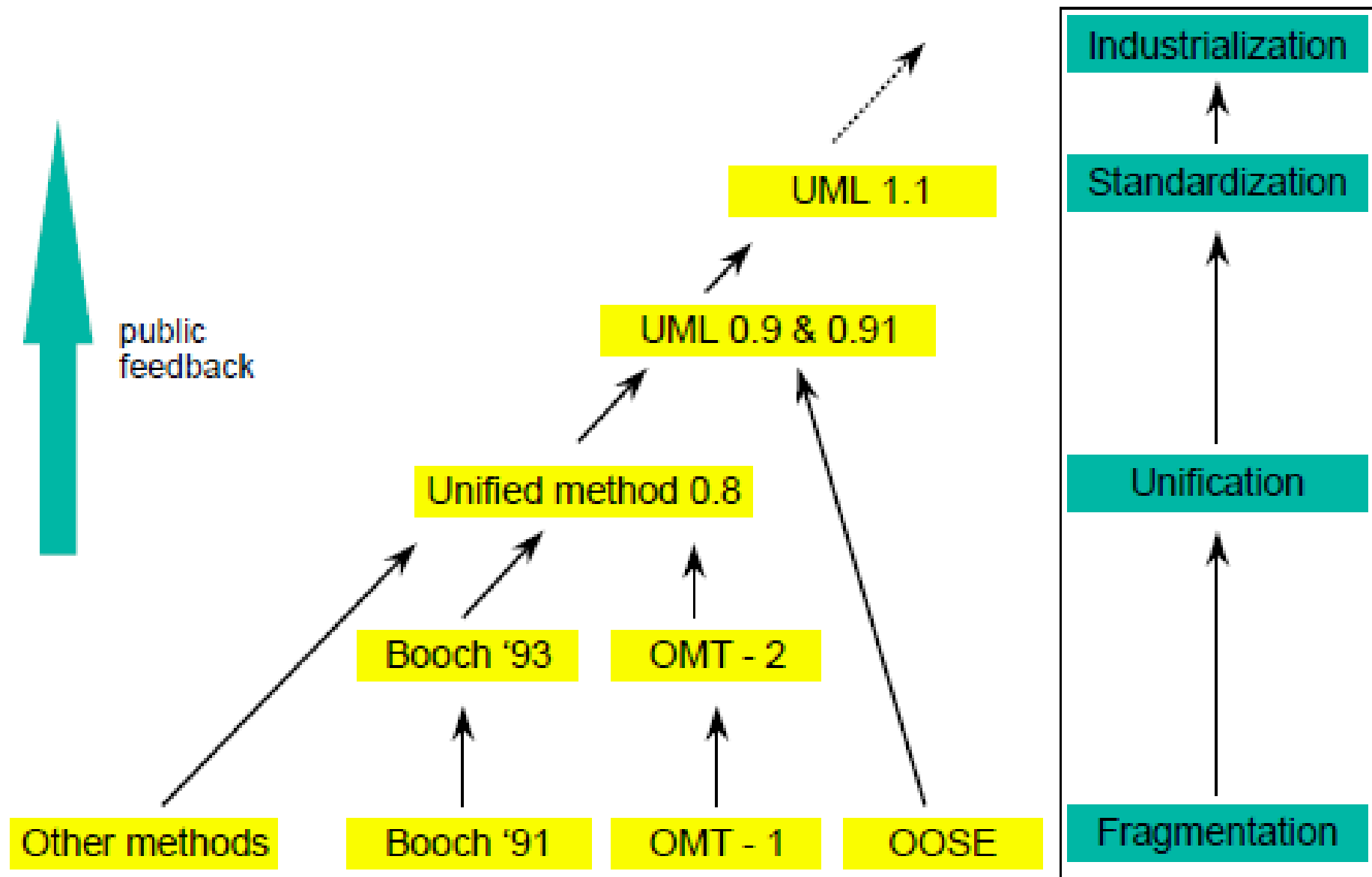
New iterations of methods appear.
Booch '93, OOSE (Jacobson), OMT-2 (Rumbaugh)

1995

Unification, UML 0.9 by Booch, Rumbaugh

1997

Standardization, UML 1.1 by Booch, Rumbaugh, Jacobson
Object Management Group (OMG) adapts UML as OOAD standard



The Unified Modeling Language

- ✓ Booch dan Rumbaugh mulai bekerja pada Unified Modeling Language (UML) pada tahun 1994 di bawah naungan Rational Inc.
- ✓ UML hanya menawarkan notasi Model, bukan metodologi bagaimana melakukan pemodelan.
- ✓ UML digunakan oleh metode pengembangan Objectory (Jacobson pada Rational).
- ✓ UML diusulkan oleh Rational Inc dan Hewlett-Packard sebagai standar untuk analisis dan desain berorientasi objek yang diadopsi oleh OMG (**Object Management Group**).
- ✓ Vendor memodifikasi alat mereka untuk membuat mereka konsisten dengan UML.

UML Diagrams (1)

○ Use Case Diagrams

Nodes:	Actor, Use (case)
Links:	Involvement, Extension, Usage

○ Class Diagrams

Nodes:	Class
Links:	Association, Generalization

○ Interaction Diagrams

Nodes:	Object
Links:	Message, Lifeline

○ State Diagrams

Nodes:	State, Sub-State
Links:	Transition

○ Activity Diagrams

Nodes:	Activity
Links:	Guard, Synchronization

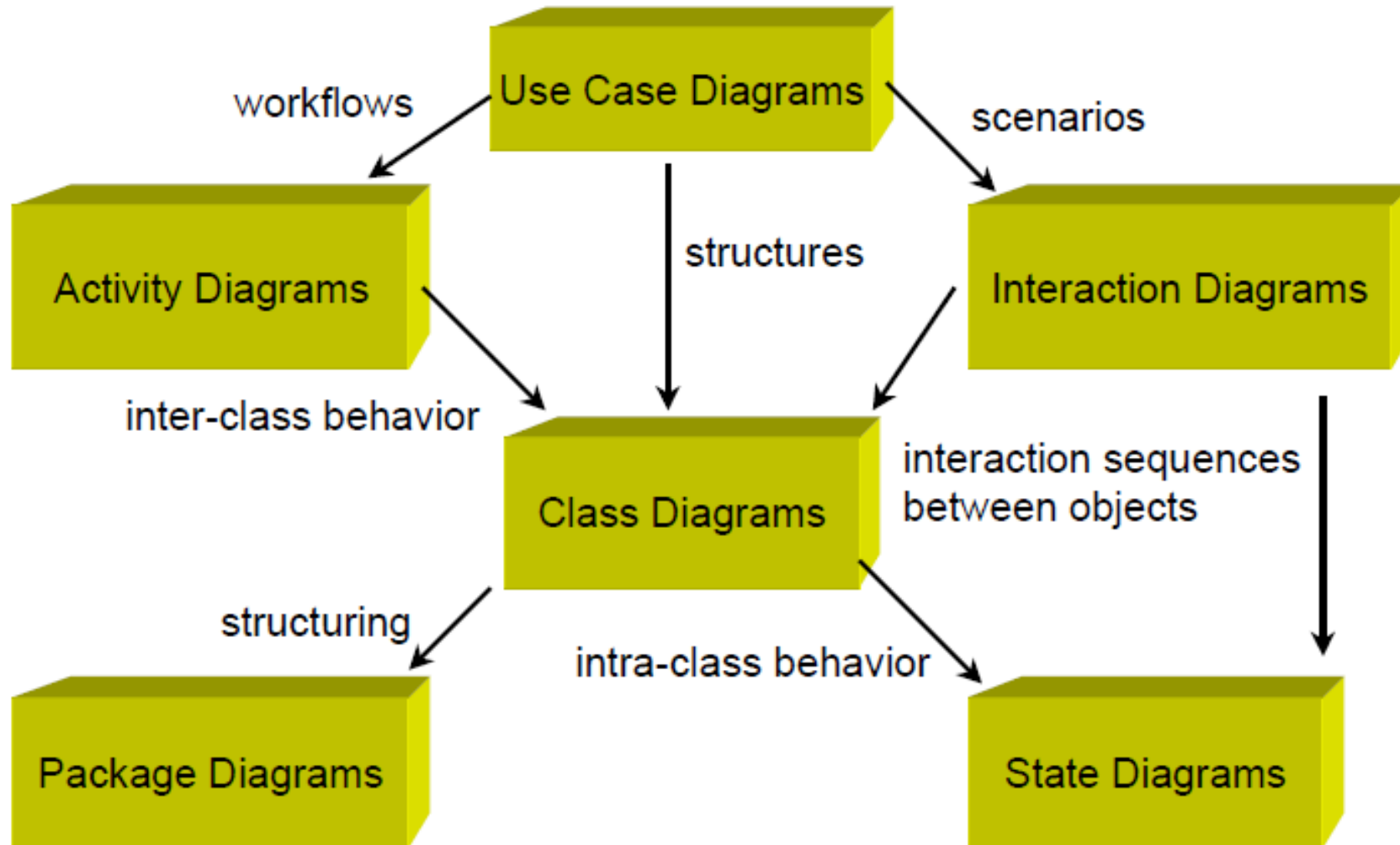
○ Package Diagrams

Nodes:	Package
Links:	Dependency

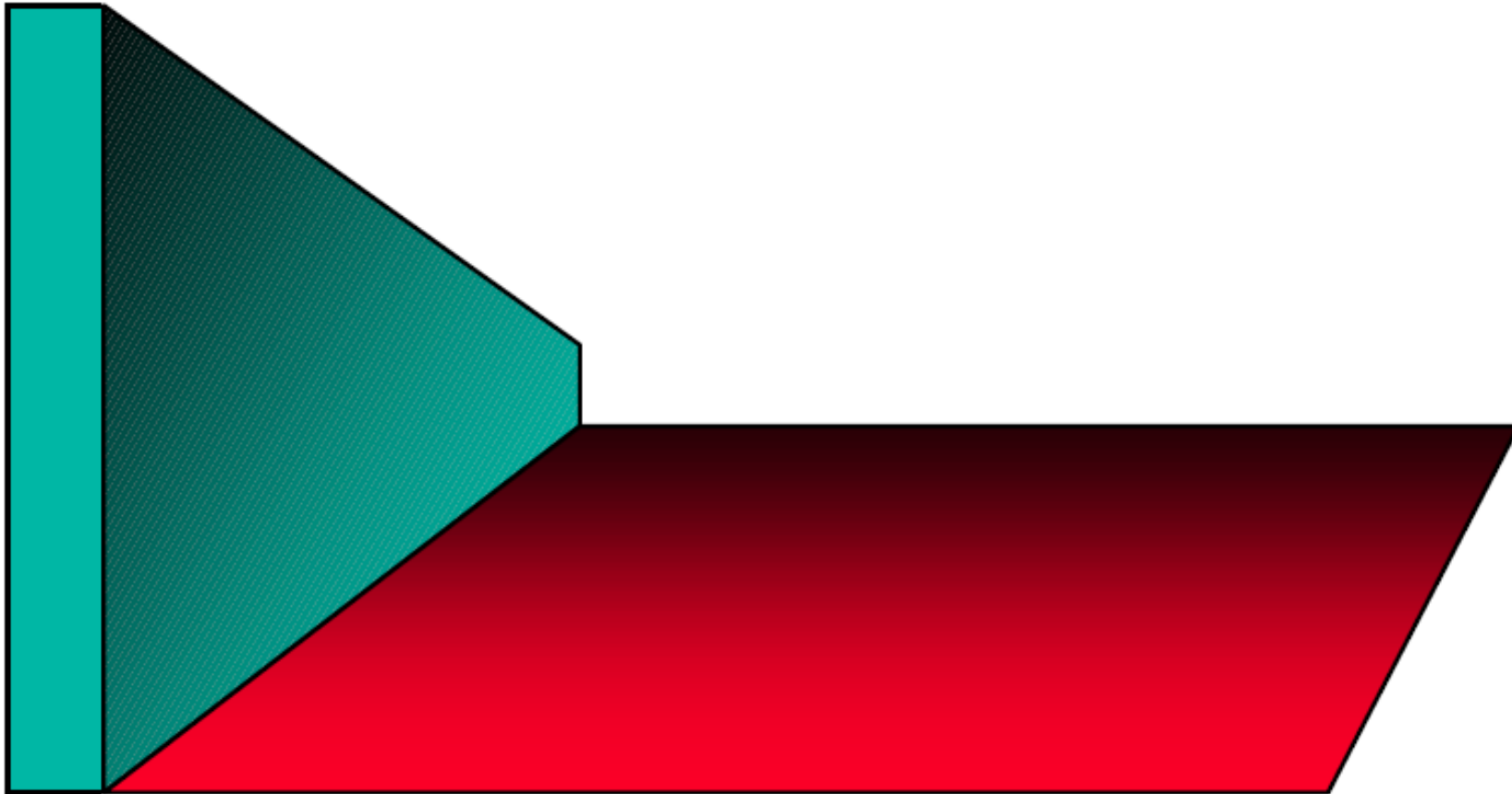
○ Deployment Diagrams

Nodes:	Processor, Node
Links:	Dependency

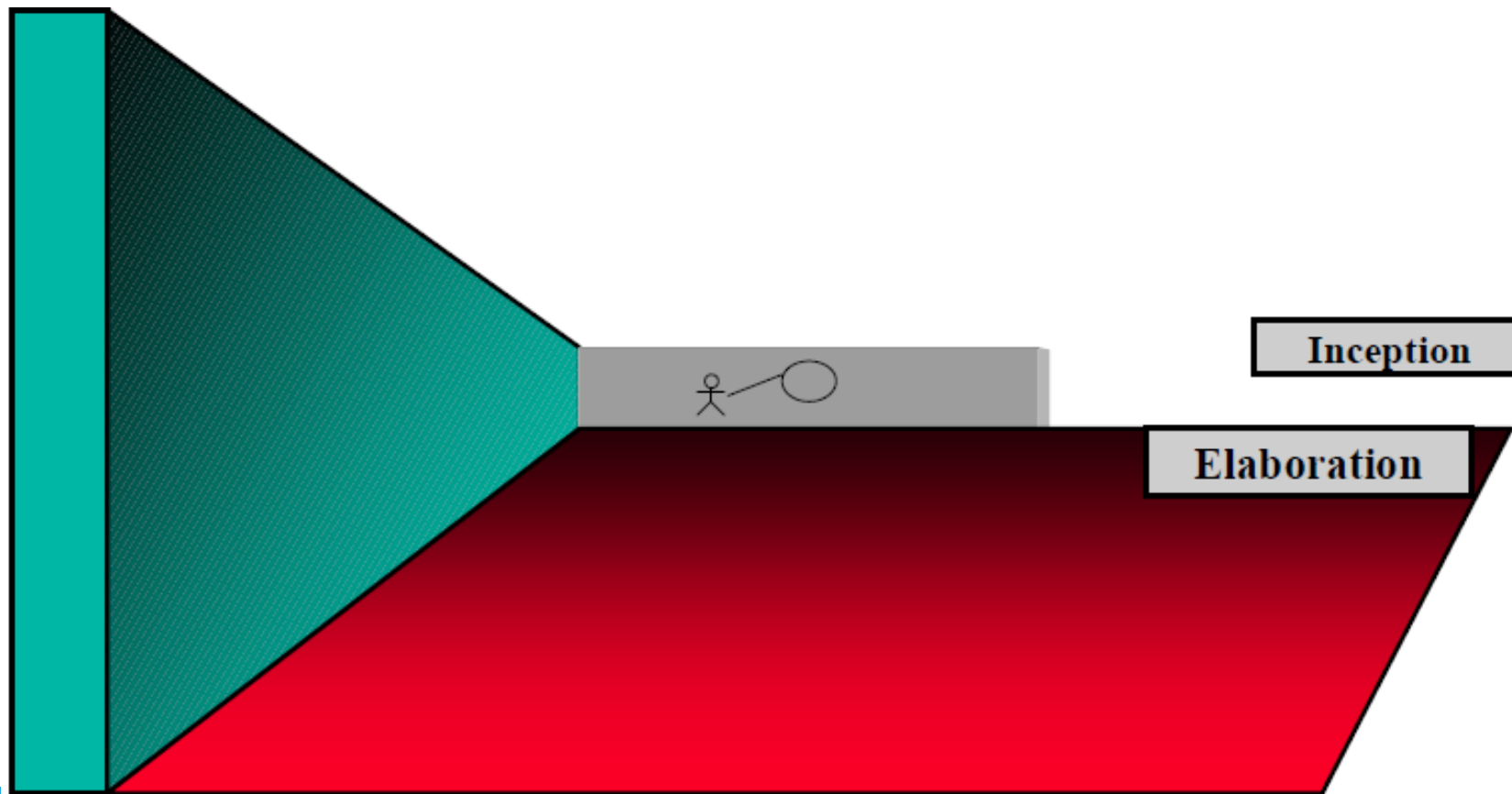
UML Diagrams (2)



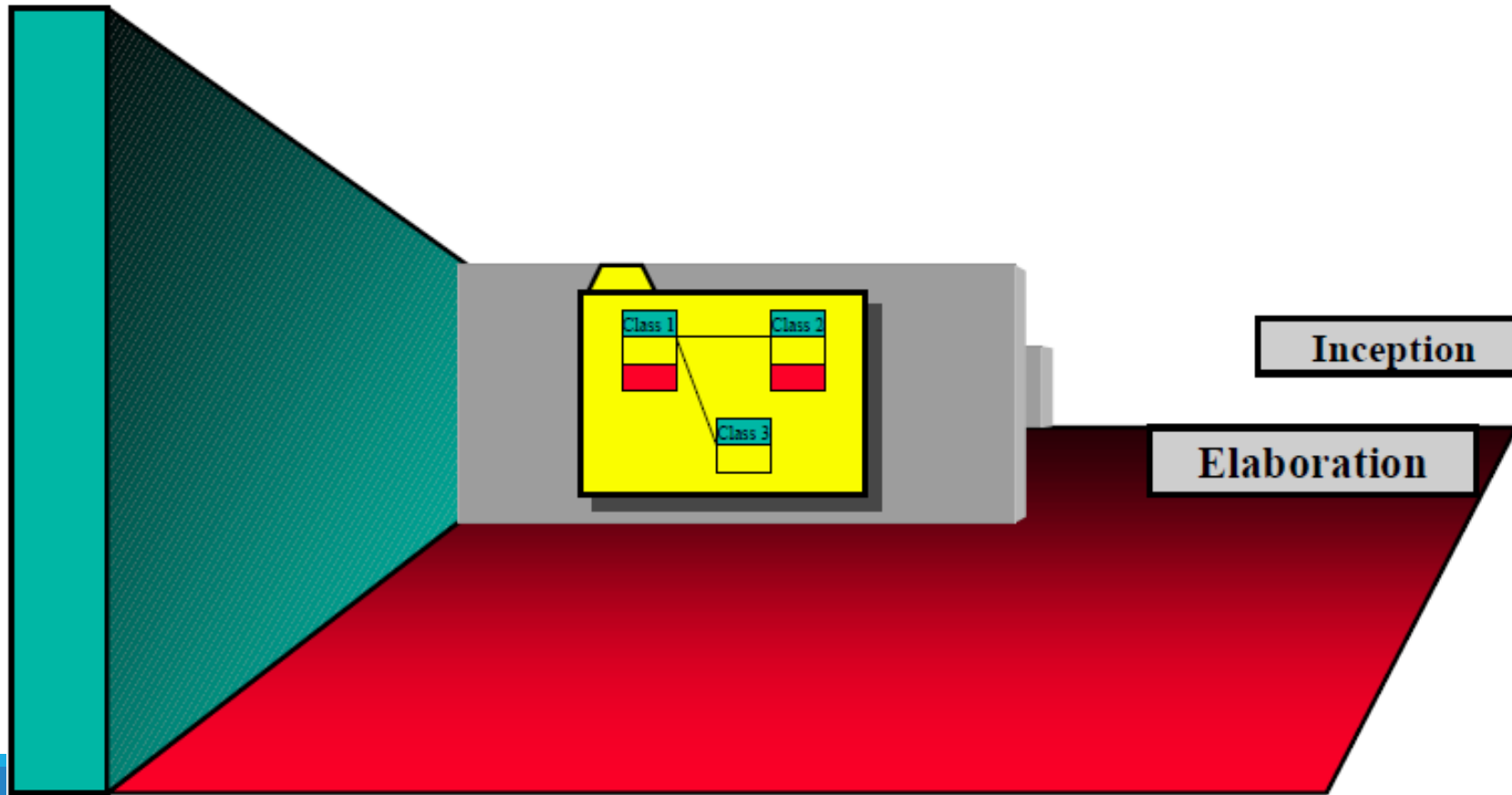
Diagrams and Process



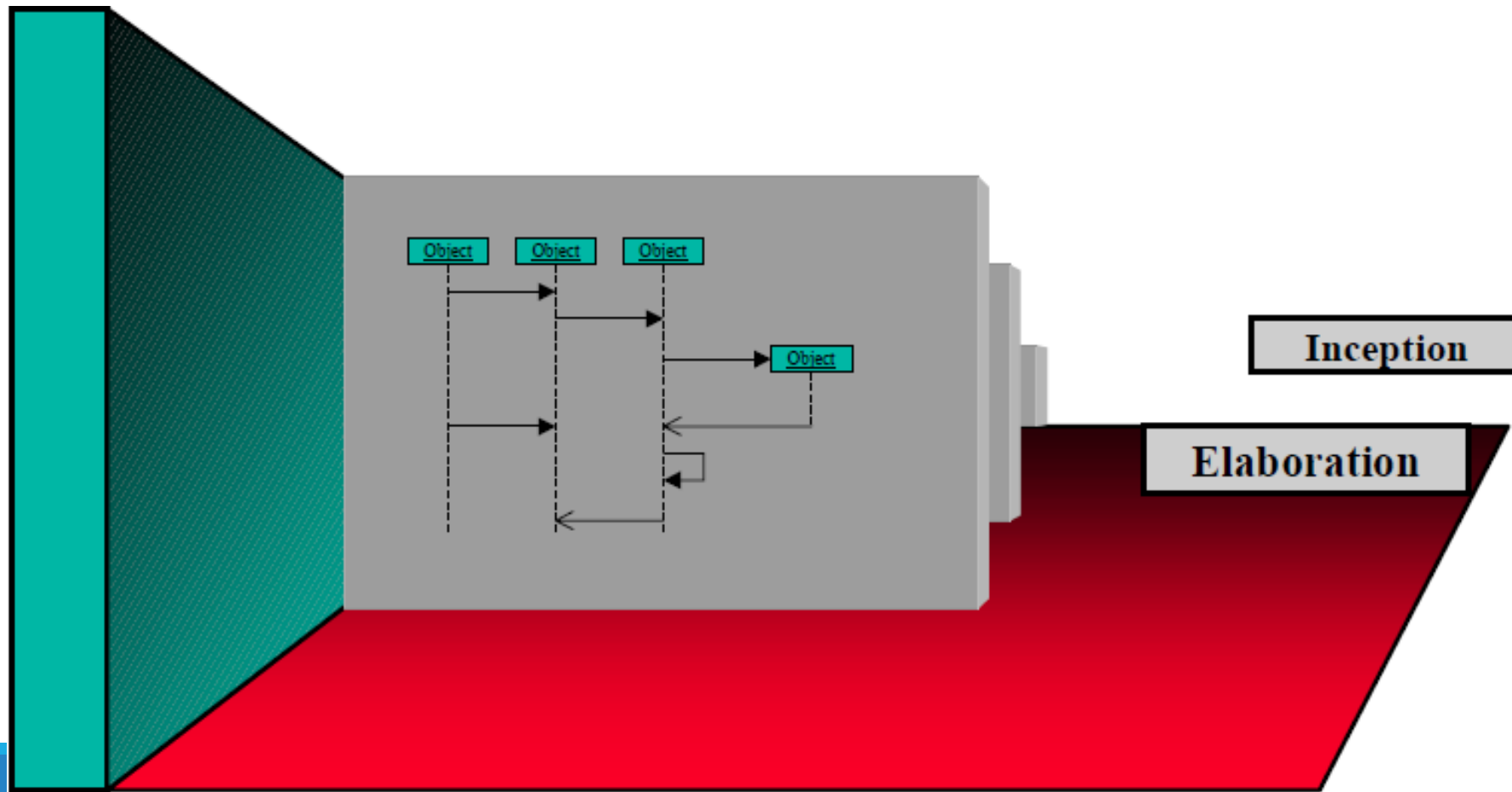
Use Case Diagrams



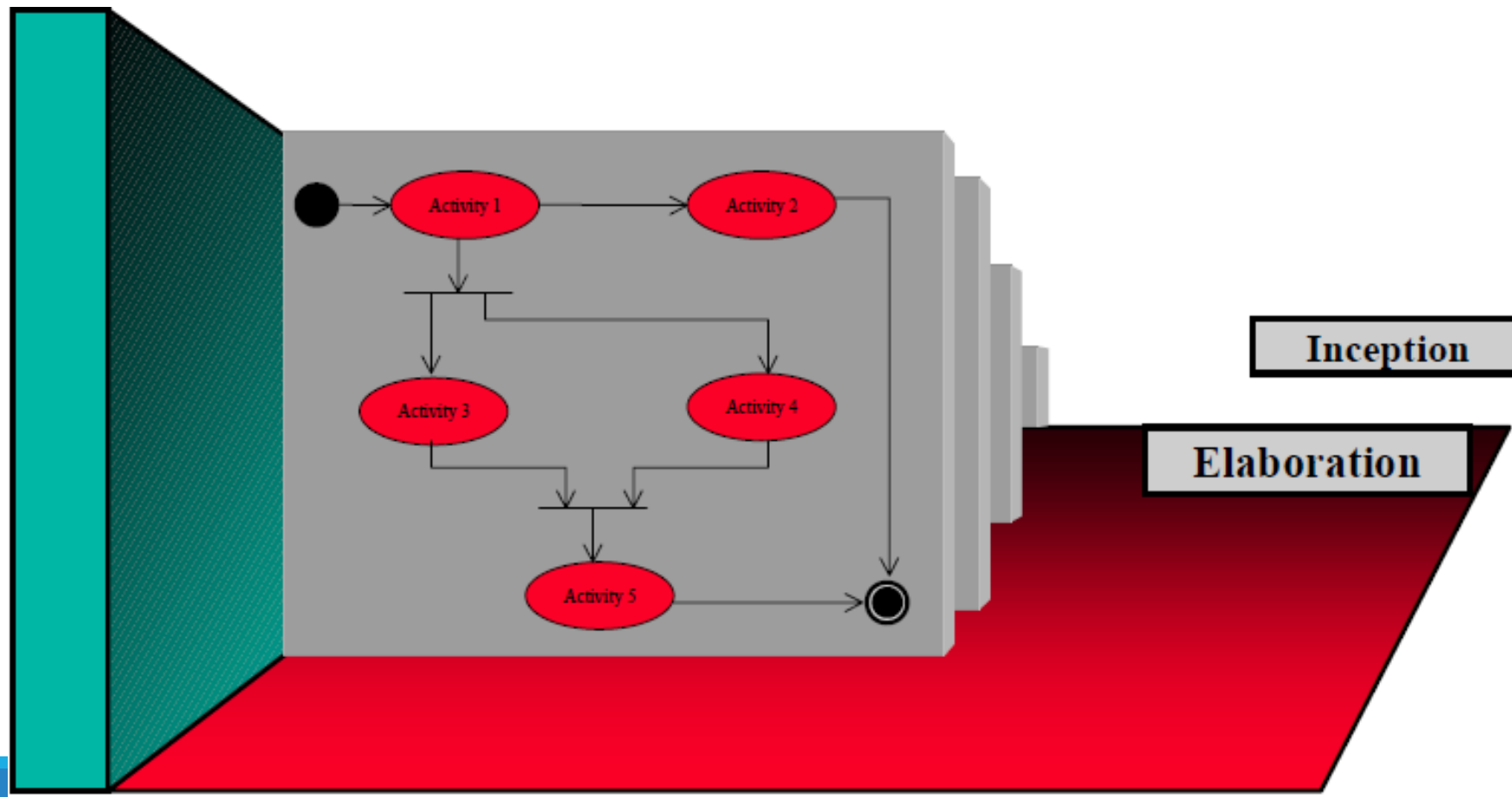
Class & Package Diagrams



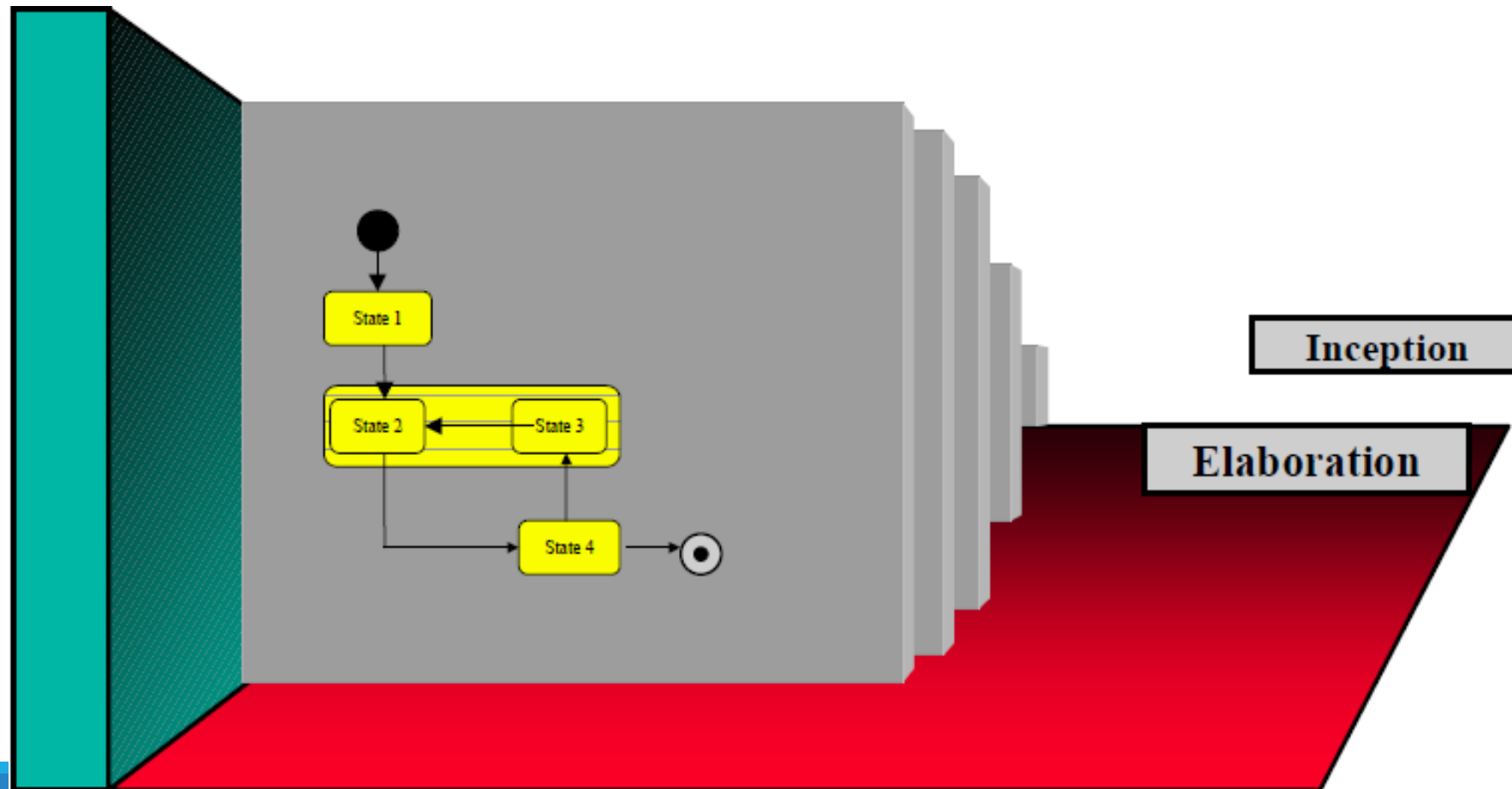
Interaction Diagrams (Scenarios)



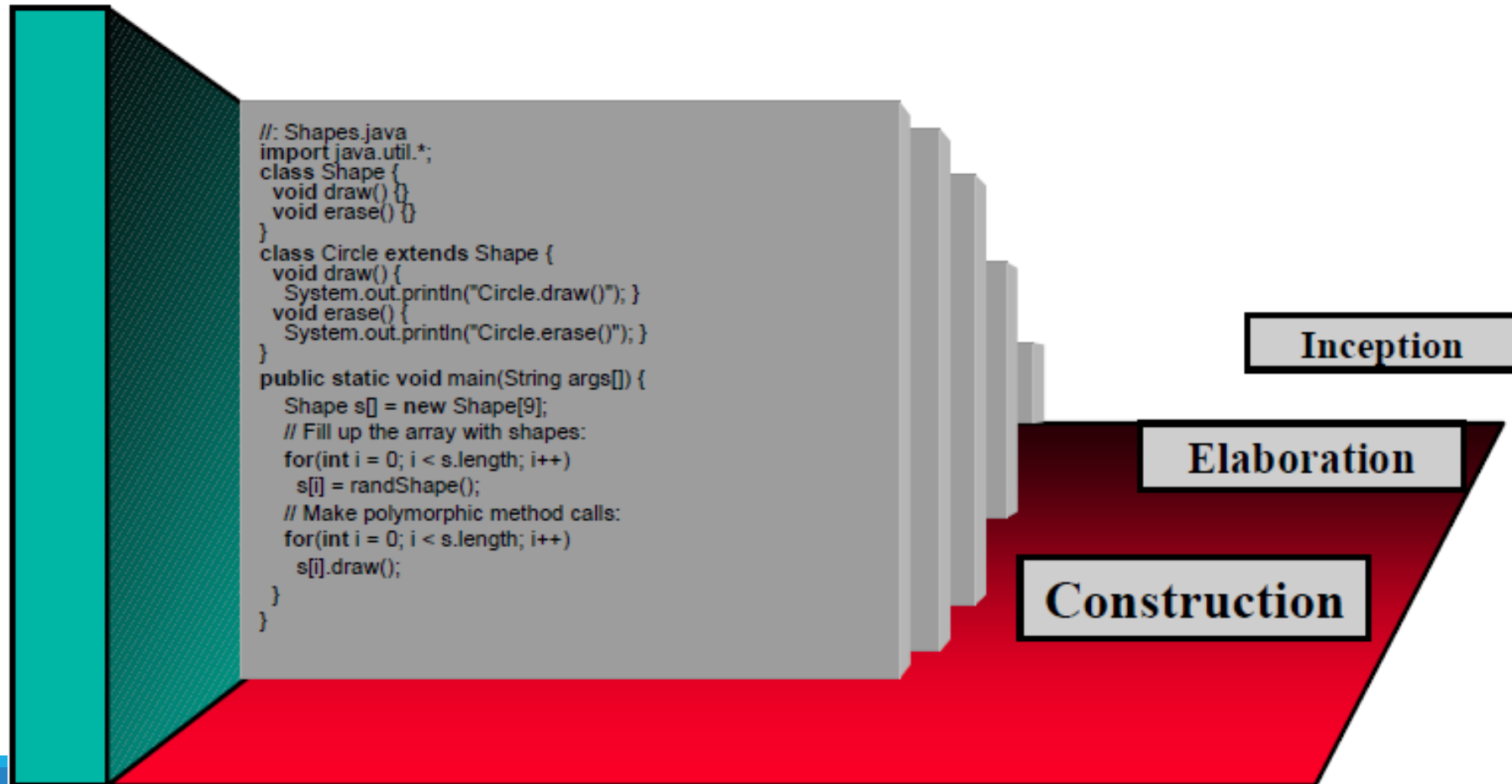
Activity Diagrams (Workflow, Interclass Behavior)



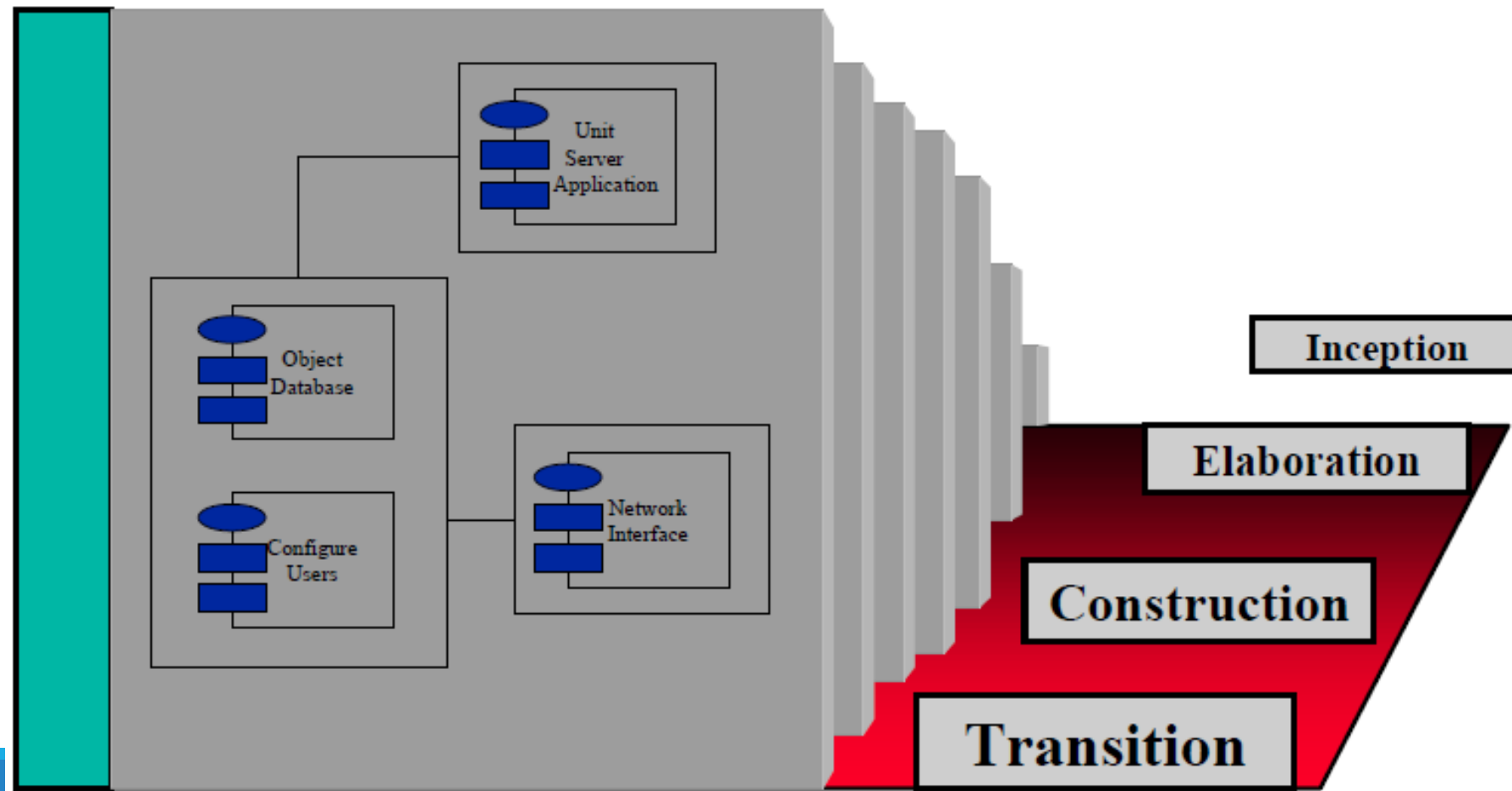
State Transition Diagrams (Intraclass Behavior)



Source Code



Deployment Diagrams



UML

Use case Diagrams

- Menggambarkan perilaku fungsional sistem seperti yang terlihat oleh penggunanya.

Class diagrams

- Menggambarkan struktur statis sistem ini: [Classes](#), [Associations](#)

Sequence diagrams

- Menggambarkan perilaku dinamis sebuah sistem: [Actors](#), [objects](#), [messages](#)

Statechart diagrams

- Menggambarkan perilaku dinamis dari objek individu dari sistem: [states](#), [events](#), [transitions](#)

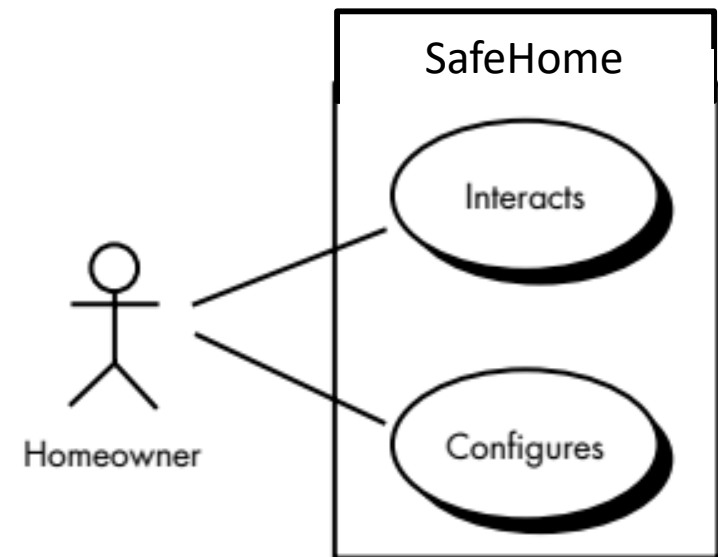
Activity Diagrams

- Memodelkan perilaku dinamis sistem ini: [activities](#), [workflows \(flowcharts\)](#)

Use Case Model

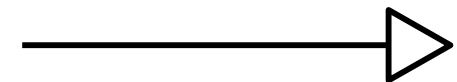
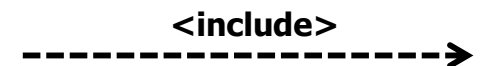
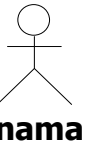
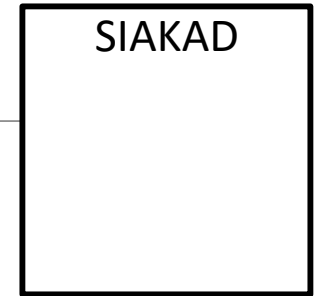
Memodelkan interaksi antara aktor dengan sistem.

Mendefinisikan kebutuhan fungsional suatu sistem.

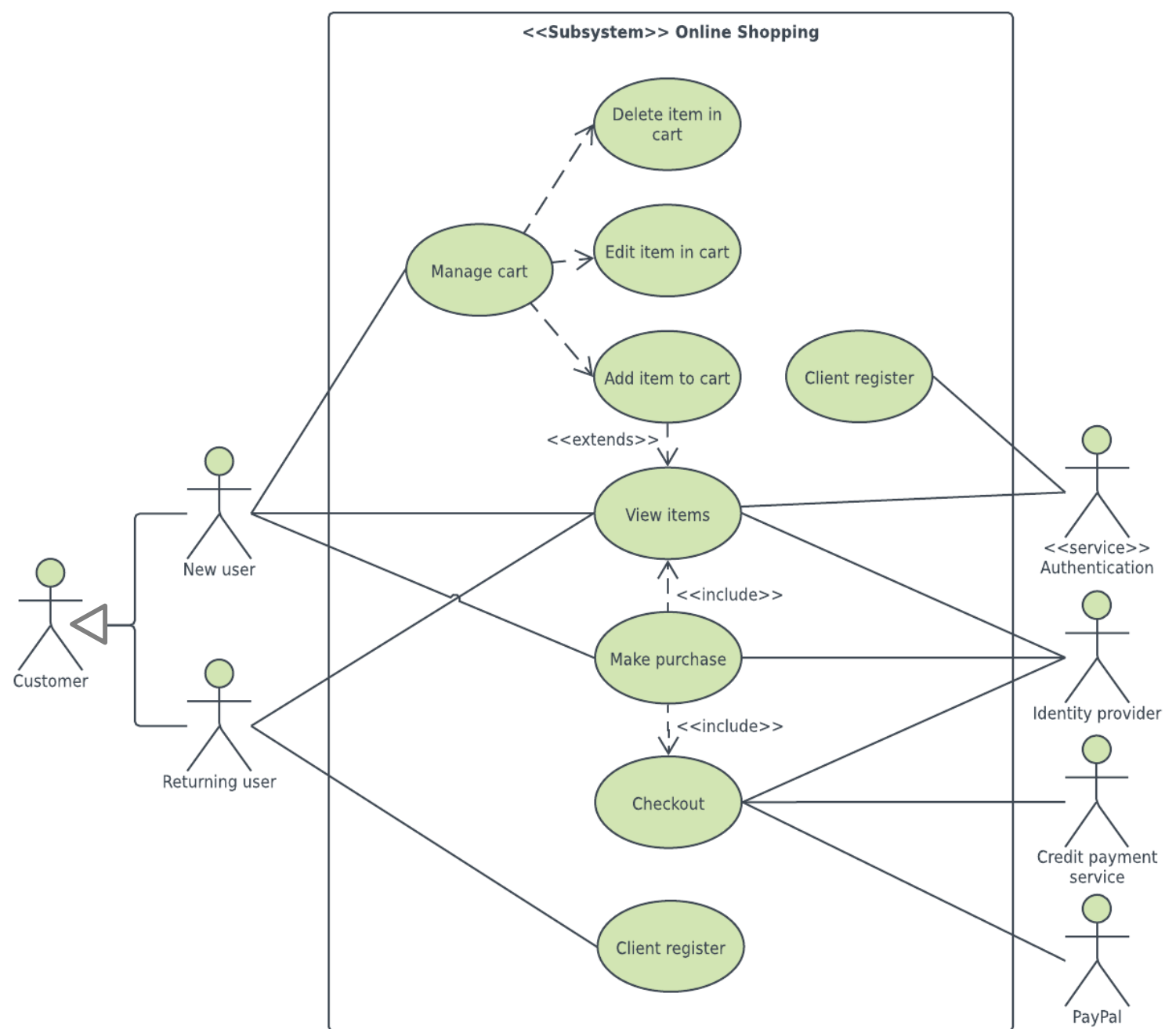


Komponen Use Case Diagram

- Boundary System : Batasan / lingkup sistem
- Aktor : Orang/sistem lain yang berinteraksi dengan sistem
- Use-cases : Fungsionalitas yang disediakan sistem
- Asosiasi : Komunikasi antara aktor dan sistem
 - Extend : relasi use case tambahan yang sifatnya opsional
 - Include : relasi use case tambahan yang sifatnya wajib dilakukan
 - Generalisasi : hubungan generalisasi-spesialisasi / umum-khusus



Contoh Use Case Diagram



Spesifikasi Use Case

Use Case - ID : UC-03

Nama Use Case : Melihat produk

Prekondisi : Customer telah login

Skenario Use Case :


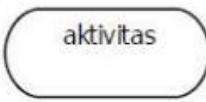



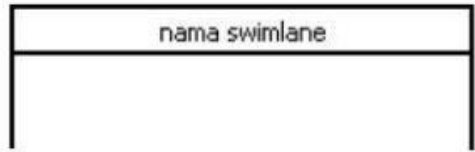
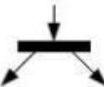

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu Daftar Produk	
	2. Menampilkan daftar produk dari basisdata ke layar
3. Menekan tombol navigasi (next, prev)	
	4. Refresh tampilan daftar produk
Skenario Alternatif	
1. Memilih menu Daftar Produk	
	2. Menampilkan pesan 'Tidak ada produk'

Activity Diagram

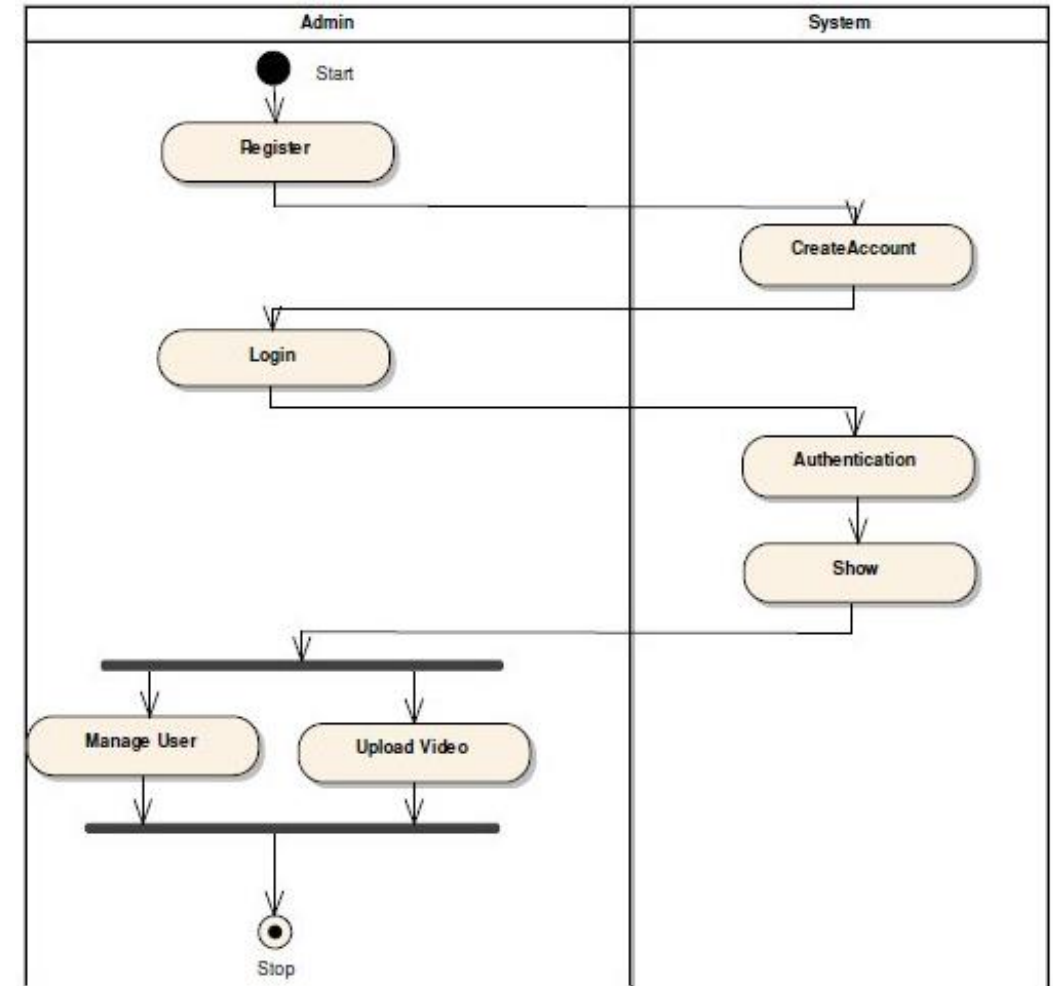
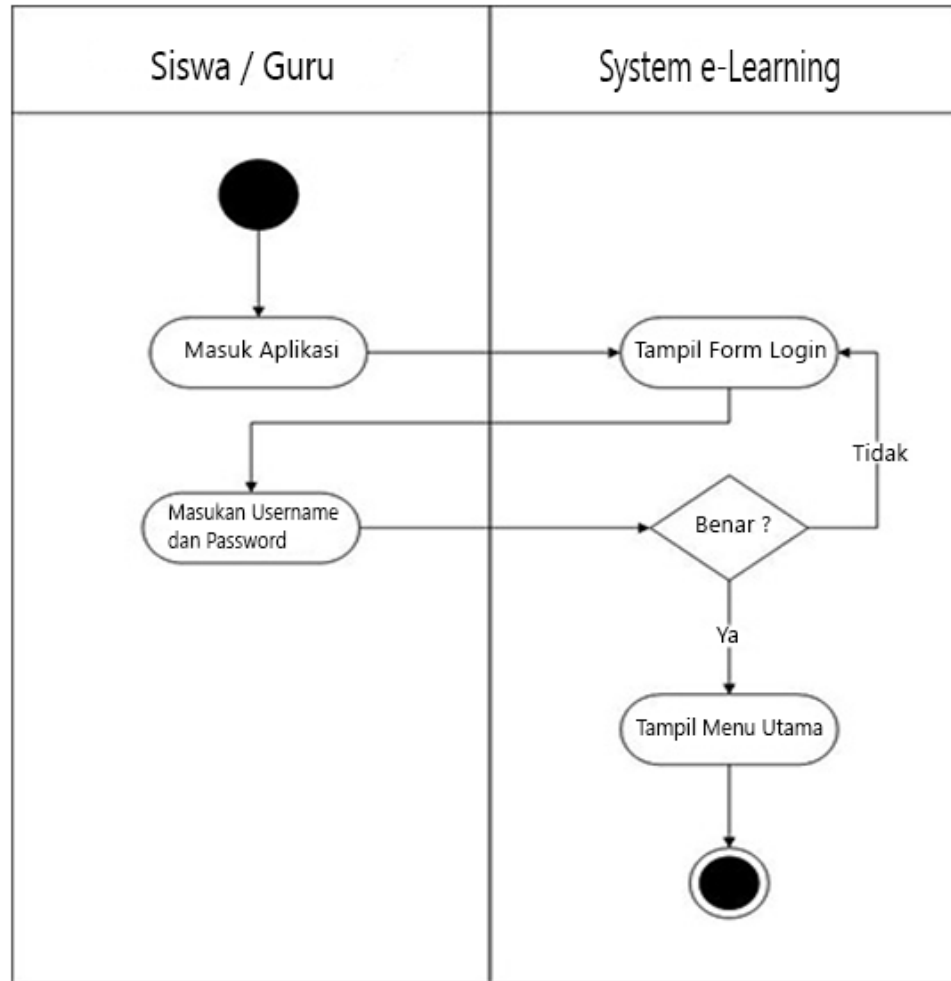
Diagram aktivitas memvisualisasikan langkah-langkah yang dilakukan dalam kasus penggunaan - aktivitas dapat berurutan, bercabang, atau bersamaan.

Jenis diagram UML ini digunakan untuk menunjukkan perilaku dinamis suatu sistem, tetapi juga dapat berguna dalam pemodelan proses bisnis.

Simbol Activity Diagram

Simbol	Deskripsi
status awal 	status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
aktivitas 	aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
percabangan / <i>decision</i> 	asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
penggabungan / <i>join</i> 	asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
status akhir 	status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
swimlane 	memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
<i>fork</i> , 	digunakan utk menunjukkan kegiatan yg dilakukan secara paralel
<i>join</i> , 	digunakan utk menunjukkan kegiatan yg digabungkan

Contoh Activity Diagram


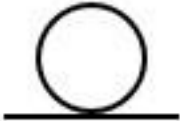
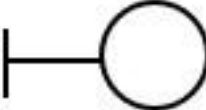





Sequence Diagram

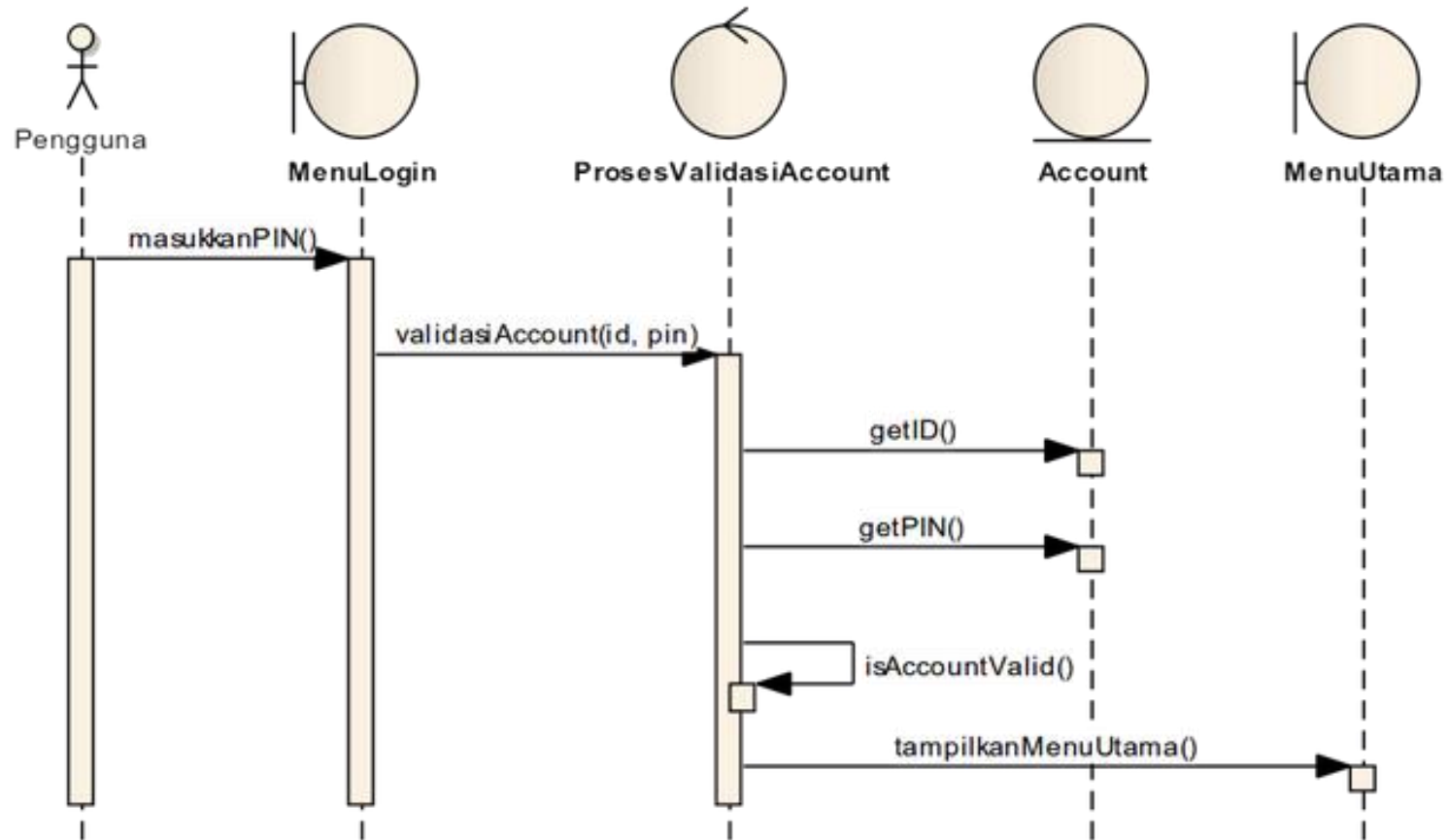
Diagram sekuens, terkadang disebut sebagai diagram peristiwa atau skenario peristiwa, menunjukkan urutan interaksi objek.

Diagram sekuens merepresentasikan skenario runtime sederhana secara visual.

Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari form
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

Contoh Sequence Diagram [1]



Contoh Sequence Diagram [2]

