Notasi *Object Oriented System*

Chapter II

Introduction

- **Unified Modeling Language (UML),** merupakan standar untuk mendokumentasikan *object-oriented systems*
- *UML is a **modeling** language, not a methodology or process*
- Istilah "*Unified*" mencerminkan bahwa UML merupakan upaya untuk menyatukan pendekatan yang berbeda.

Modeling

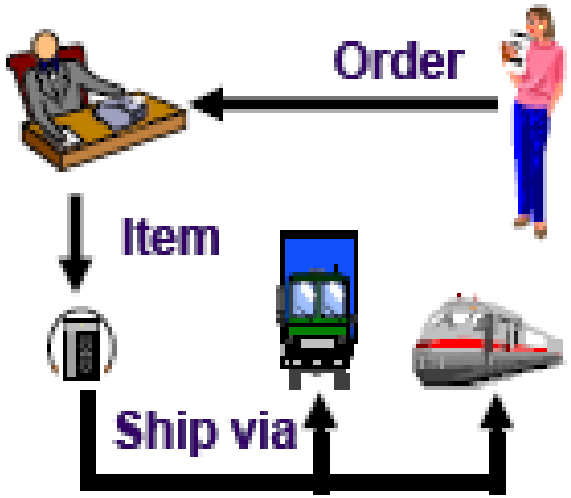
Why Modeling?

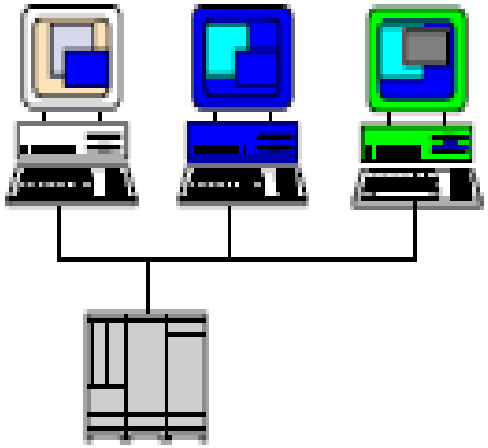
- ***Analyse the problem-domain* (Analisa Domain Permasalahan)**
  - *simplify reality* (menyederhanakan realita)
  - *capture requirements* (menangkap kebutuhan)
  - *visualize the system in its entirety* (menggambarkan sistem secara keseluruhan)
  - *specify the structure and/or behaviour of the system* (menentukan struktur dan atau perilaku sistem)

- ***Design the solution* (Merancang Solusi)**
  - *document the solution - in terms of its structure, behaviour, etc* (mendokumentasikan solusi, dalam hal struktur, perilaku,dll)

Modeling (1)

Why Modeling?





Business Process

Computer System

Modeling menangkap bagian penting dari sistem
(James Rumbaugh)

Modeling (2)

Principles of Modeling

- *Choose your model well* (Pilih model Anda dengan baik)
  - Pilih model berampak pada analisis masalah dan desain solusi
- *Every model may be expressed at different levels of precision* (Setiap model dapat dinyatakan pada tingkat presisi yang berbeda)
  - Model yang sama dapat ditingkatkan untuk presisi yang berbeda.
- *The best models are connected to reality* (Model terbaik adalah yang terhubung dengan realitas) –
  - Simplify the model, but don't hide important details.
- No single model suffices (Tidak ada model tunggal sudah cukup)
  - Setiap sistem memiliki dimensi yang berbeda dengan masalah dan solusinya.

What's UML?

- UML dapat digunakan untuk memodelkan semua proses dalam siklus hidup pengembangan (*development life cvcle*) dan seluruh teknologi implementasi yang berbeda

- UML adalah suatu bahasa pemodelan untuk memvisualisasikan,menspesifiksi, konstruksi, dan mendokumentasikan artifak dari sistem perangkat lunak

- UML adalah suatu alat komunikasi untuk team dan para *stakeholders*

What's UML?

UML is a modeling language for visualising, specifying, constructing and documenting the artifacts of software systems.

Visualising - *a picture is worth a thousand words; a graphical notation articulates and unambiguously communicates the overall view of the system (problem-domain).*

Suatu gambar bernilai seribu kata kata; notasi grafis mengartikulasikan dan jelas mengkomunikasikan pandangan keseluruhan sistem (masalah-domain).

What's UML?

Specifying - *UML provides the means to model precisely, unambiguously and completely, the system in question.*

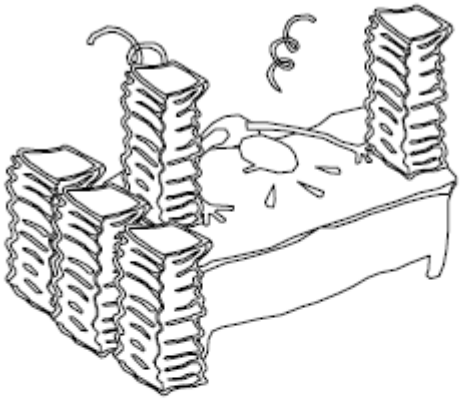
- UML menyediakan cara untuk memodelkan secara tepat, jelas dan lengkap, yang sistem pertanyakan.

What's UML?

Constructing - *models built with UML have a "design" dimension to it; these are language independent and can be implemented in any programming language.*

- Model yang dibangun dengan UML memiliki dimensi "desain" untuk dapat diimplementasikan dalam bahasa pemrograman

What's UML?



Documenting - *every software project involves a lot of documentation - from the inception phase to the deliverables.*

Documentation is (among others) for:

- Requirements
- Design
- Tests

UML provides the notations for documenting some of these artifacts

- Setiap proyek perangkat lunak melibatkan banyak dokumentasi dari fase awal sampai pengiriman

History of UML

- **Pada Oktober 1994**, Dr. James Rumbaugh bergabung dengan Perusahaan Rational sotware, dimana Grady Booch sudah bekerja disana sebelumnya.

- Grady Booch mengembangkan *Object Oriented Design* (OOD) dan Dr. James Rumbaugh mengembangkan *Object Modeling Technique* (OMT)

- Oktober 1995 menghasilkan Unified Method versi 0.8.

History of UML (1)

- Musim gugur 1995 Dr. Ivar Jacobson ikut pula bergabung dengan duet Rumbaugh-Booch, dengan memperkenalkan *tool use case*.
- Trio tersebut pada bulan Juni 1996 menghasilkan Unified Modeling Language (UML) versi 0.9
- Sebelumnya Dr. Ivar Jacobson mengembangkan *Object Oriented Software Engineering* (OOSE). Trio ini mengembangkan Ratinal Unified Process (RUP)

History of UML (2)

- Banyak perusahaan software merasakan bagaimana pentingnya UML dalam tujuan strategis. Beberapa perusahaan membentuk sebuah konsorsium yang terdiri dari perusahaan-perusahaan:
  - Microsoft
  - Oracle
  - IBM
  - Hewlett-Packard
  - Intellicorp
  - I-Logix
  - DEC, Digital Equipment Corp
  - Texas instrumen

History of UML (3)

- Dari konsorsium tersebut pada bulan Januari 1997 lahirlah UML versi 1.0

- Pada bulan September 1997 lahirlah UML versi 1.1, dengan 8 buah diagram: 1) Use Case Diagram, 2)Activity Diagram, 3) Sequence Diagram, 4) Collaboration diagram, 5) Class diagram, 6) Statechart diagram, 7) Component diagram, 8) Deployment diagram

History of UML (4)

- Pada bulan November 1997 sebuah organisasi non profit standarisasi *Object Management Group* (OMG) mengakui UML sebagai sebuah bahasa pemodelan standar untuk aplikasi *object oriented*
- OMG didirikan pada bulan April 1989 dengan kantor pusat di Needham, MA, USA. (www.omg.org)
- Pada tahun 1999 lahirlah UML versi 1.3, menjadi 9 buah diagram, dengan penambahan: *Business use case Diagram*

History of UML (5)

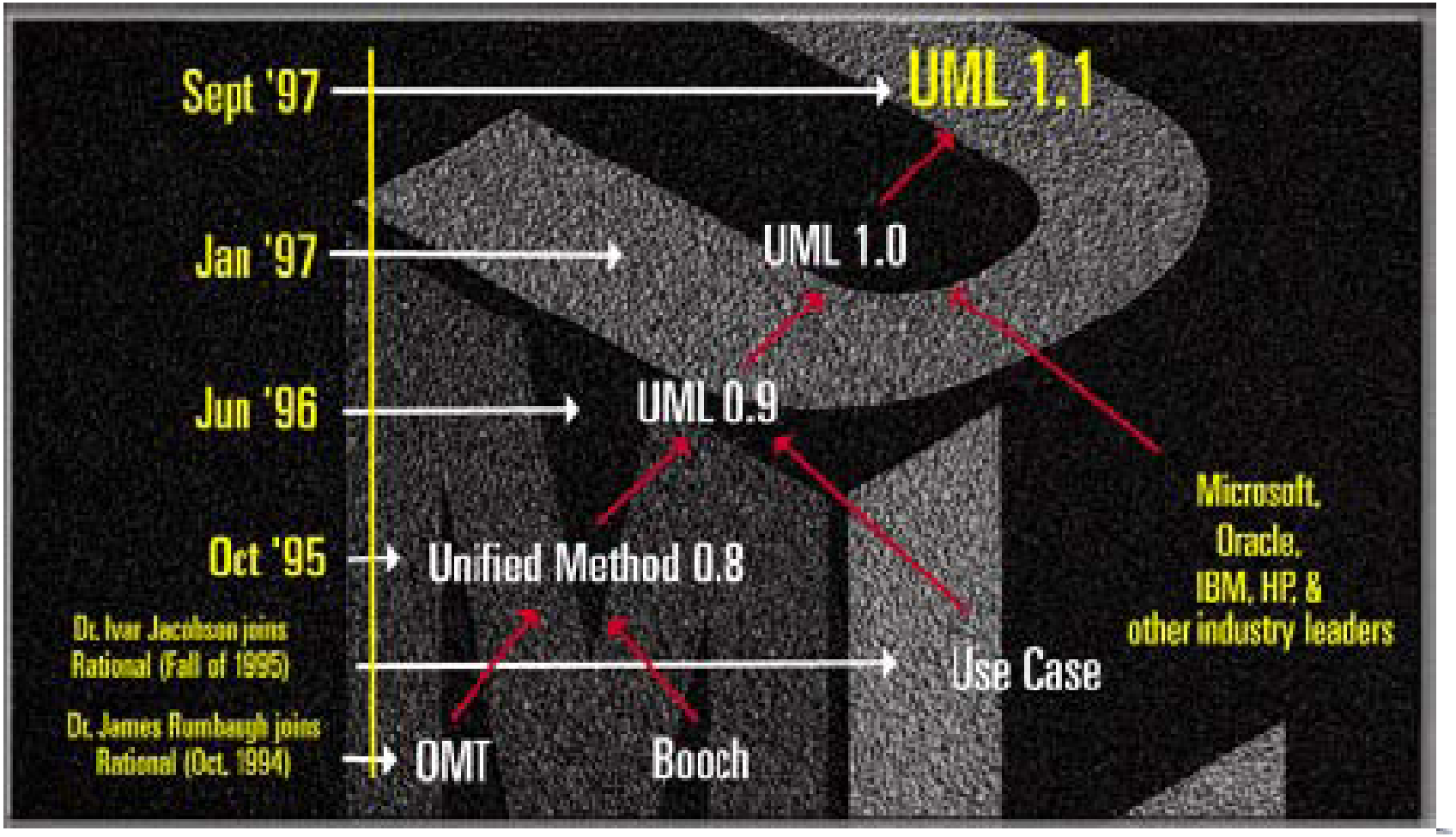
- Pada May 2001 lahirlah UML versi 1.4, menjadi 10 buah diagram, dengan penambahan *Object Diagram*

- Pada tahun 2002 lahirlah UML versi 2.0, menjadi 13 buah diagram, dengan penambahan dan penggantian yaitu:
- 1. Use Case Diagram
- 2. Activity Diagram
- 3. Sequence Diagram

History of UML (5)

4. Communication Diagram (Collaboration diagram versi 1.x)
5. Class Diagram
6. State Machine Diagram (Statechart diagram versi 1.x)
7. Component Diagram
8. Deployment Diagram
9. Composite Structure Diagram
10. Interaction Overview Diagram
11. Object Diagram
12. Package Diagram
13. Timing Diagram

History of UML (6)



Object-oriented Systems

- UML menyediakan notasi bergambar atau grafis untuk mendokumentasikan artefak seperti kelas, objek dan paket yang membentuk sistem berorientasi objek. Diagram UML dapat dibagi menjadi tiga kategori

- *1. Structure diagrams*
- *2. Behaviour diagrams*
- *3. Interaction diagrams*

Object-oriented Systems

1. **Structure diagrams:** menunjukkan arsitektur statis dari sistem terlepas dari waktu. Sebagai contoh, diagram struktur untuk sistem universitas mungkin termasuk diagram yang menggambarkan desain kelas seperti mahasiswa, Fakultas, dll

2. **Behaviour diagrams:** yang menggambarkan perilaku sistem atau proses bisnis. Untuk sistem universitas, diagram perilaku yang mungkin akan menunjukkan bagaimana siswa mendaftar untuk mengikuti perkuliahan.

3. **Interaction diagrams:** menunjukkan metode, interaksi dan kegiatan objek.

Structure diagrams

Termasuk pada *Structure Diagram* meliputi:

1. *Class diagrams*
2. *Composite structure diagrams*
3. *Component diagrams*
4. *Deployment diagrams*
5. *Object diagrams*
6. *Package diagrams*

Structure diagrams

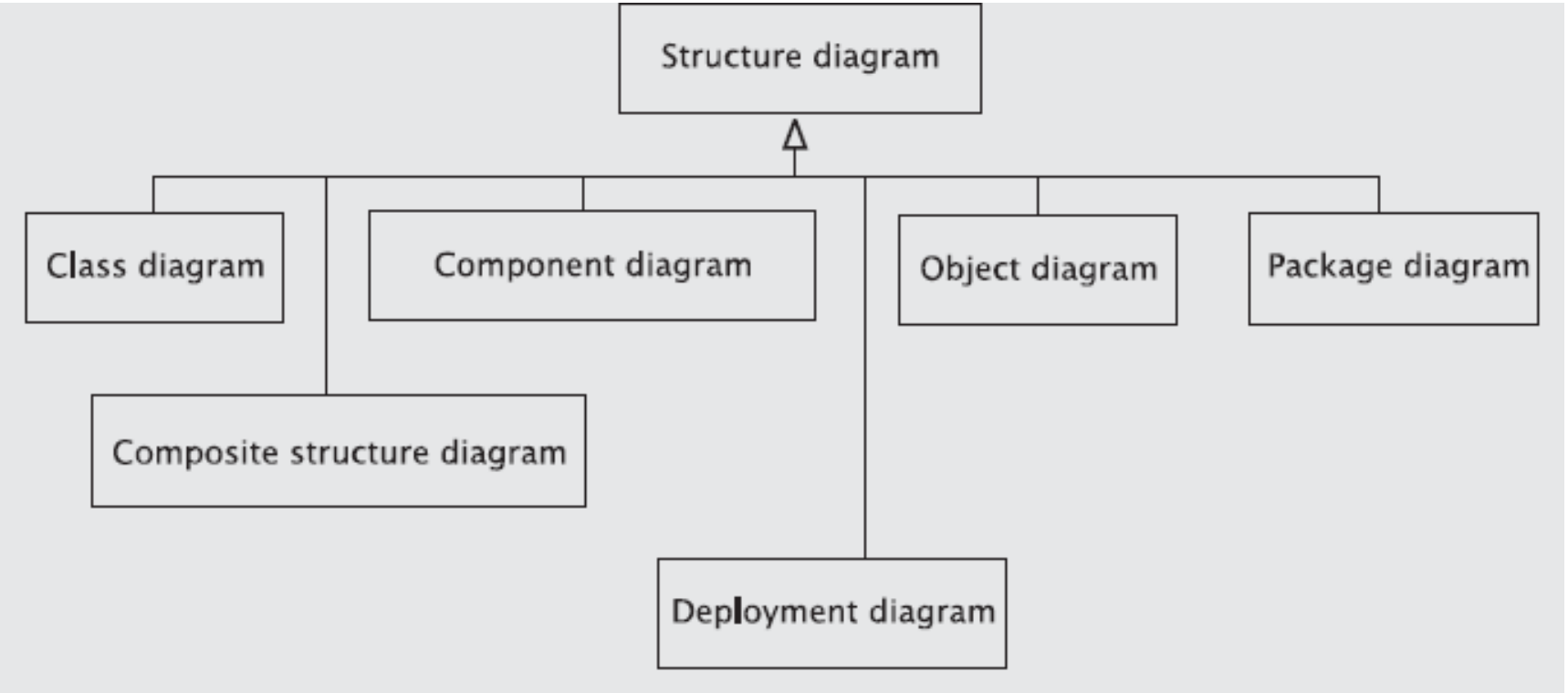


Figure 2.2 *Types of UML structure diagrams*

Behaviour diagrams

Termasuk pada *Behaviour diagrams* meliputi:

- *1. Activity diagrams*
- *2. Use case diagrams*
- *3. State machine diagrams*

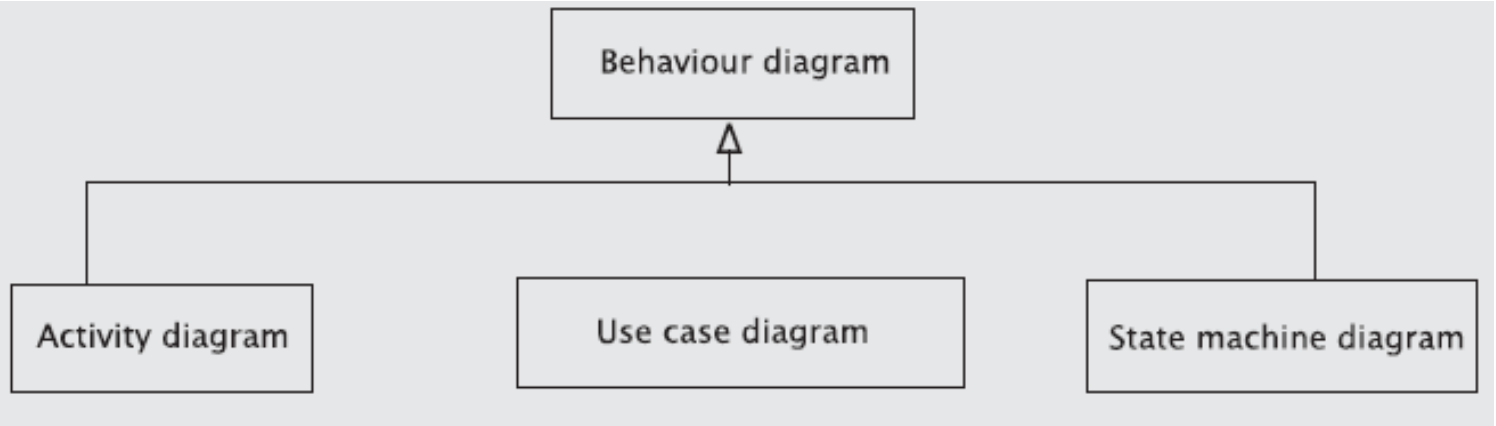


Figure 2.3 *Types of UML behaviour diagrams*

Interaction diagrams

Termasuk pada *Interaction diagrams* meliputi:

- *1. Sequence diagrams*
- *2.Timing diagrams*
- *3. Communication diagrams*
- *4. Interaction overview diagrams*

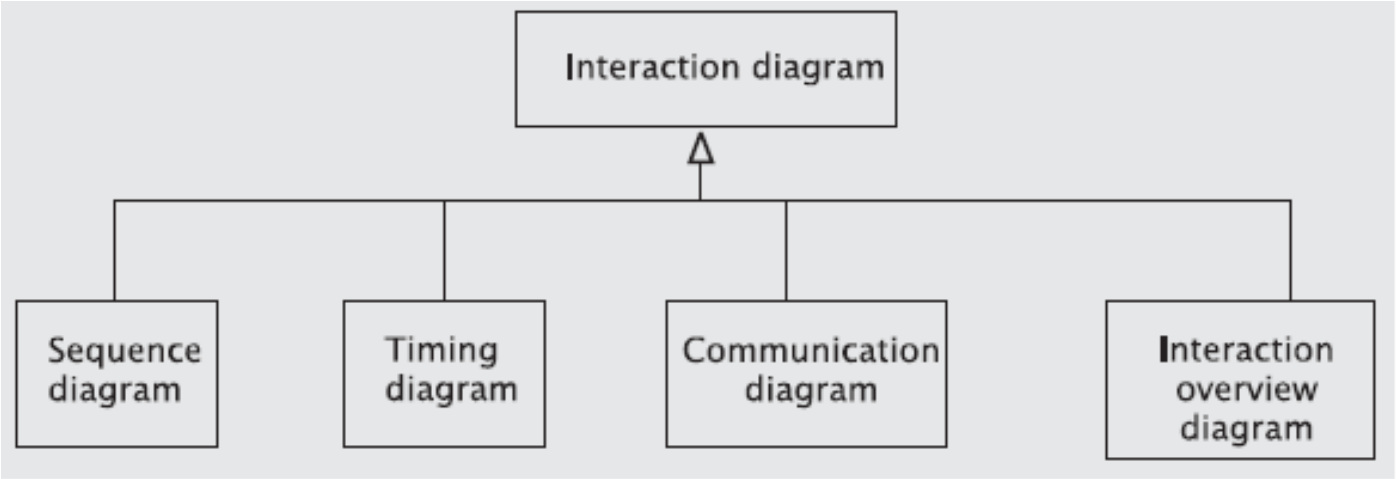
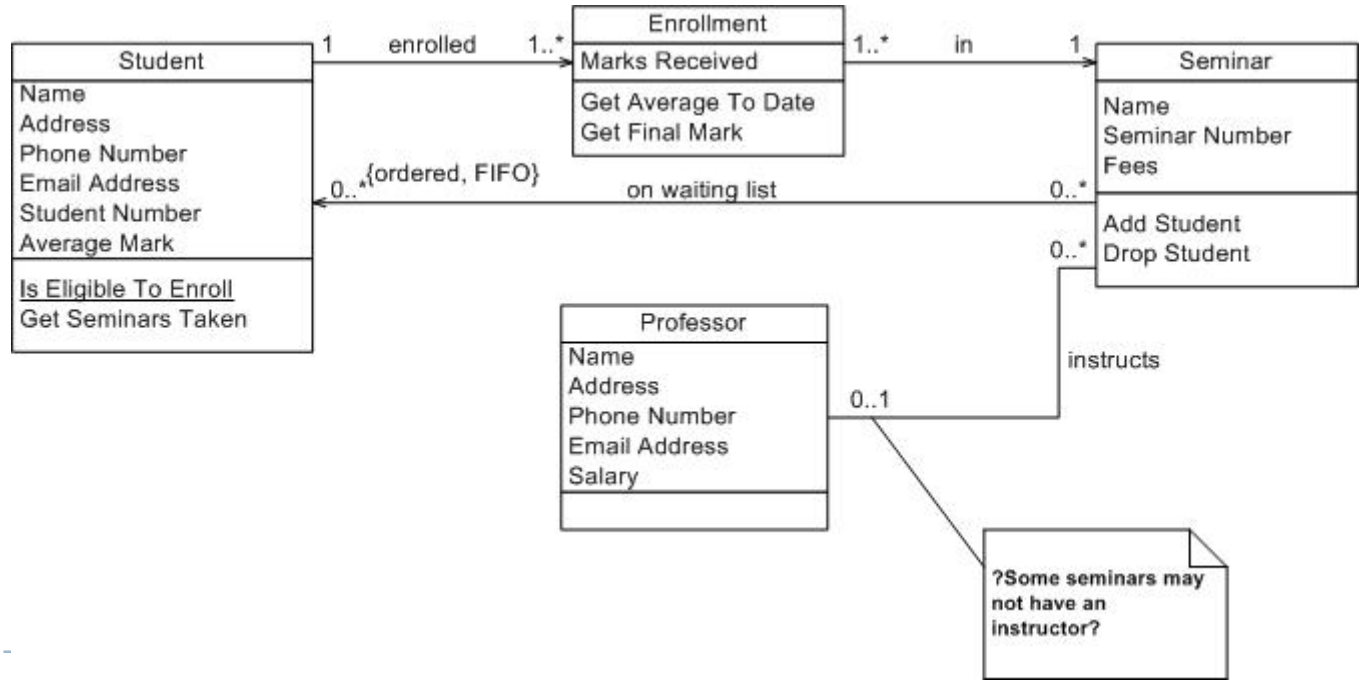


Figure 2.4 *Types of UML interaction diagrams*

UML Diagrams

Class Diagram

- Digunakan untuk mengilustrasikan hubungan antara kelas-kelas pada sistem
- Mewakili sesuatu/benda (*employee*, *paycheck*,..)



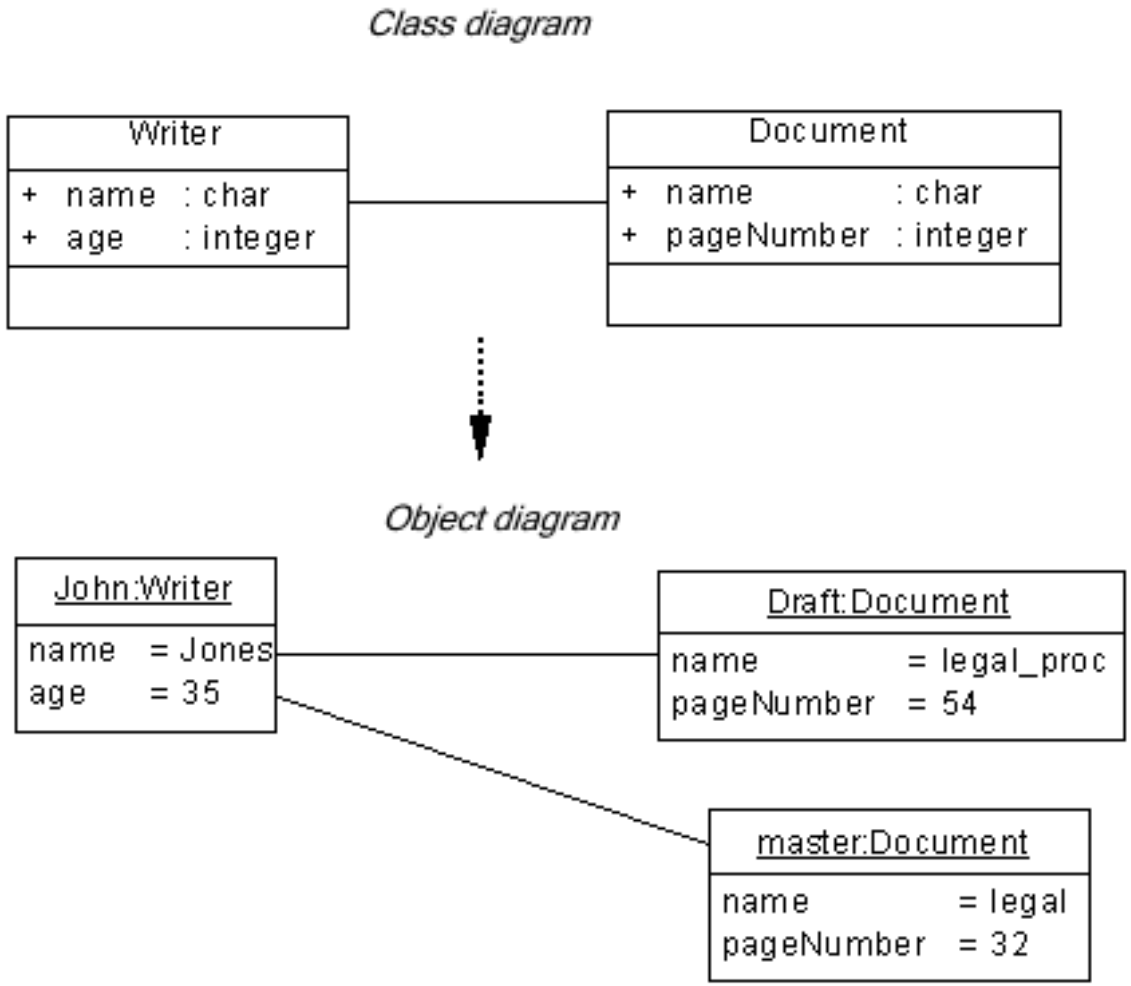
UML Diagrams

Object Diagram

- Mirip dengan *Class Diagram*
- Gambaran tentang objek-objek dalam sistem
- Hubungan antar objek

UML Diagrams

- The difference between Class Diagram and Object Diagram



UML Diagrams

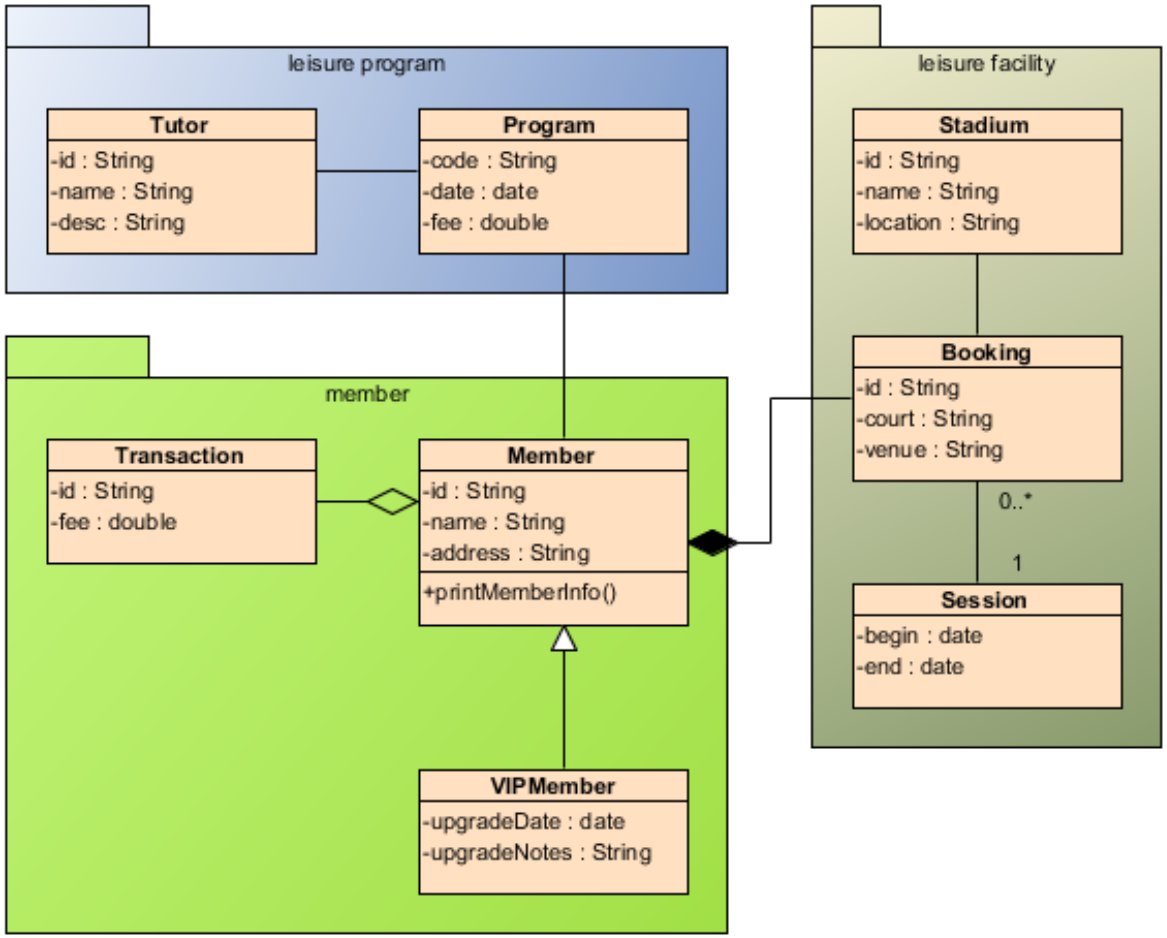
Package Diagram

- Sebuah *package* adalah sebuah bentuk pengelompokan yang memungkinkan untuk mengambil setiap bentuk di UML dan mengelompokkan elemen-elemennya dalam tingkatan unit yang lebih tinggi

- Kegunaan paling umum adalah untuk mengelompokkan *class*

UML Diagrams

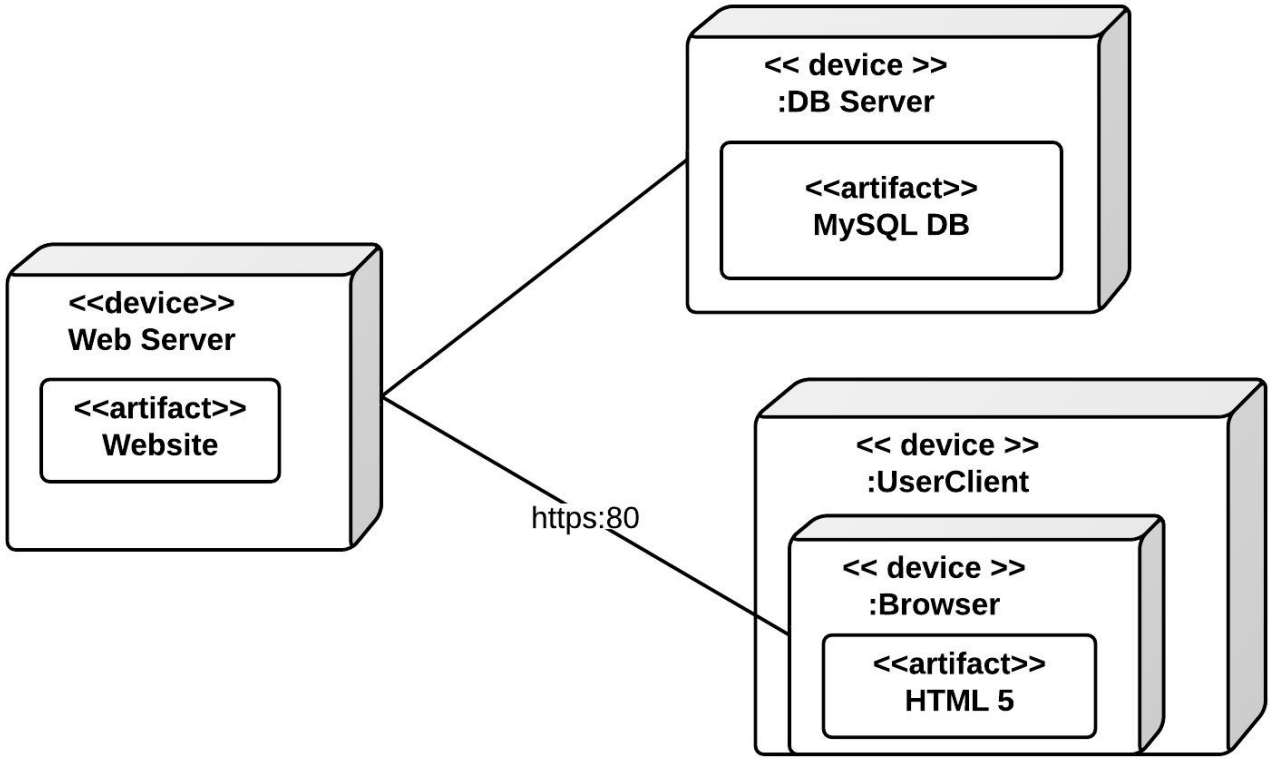
Package Diagram



UML Diagrams

Deployment Diagram

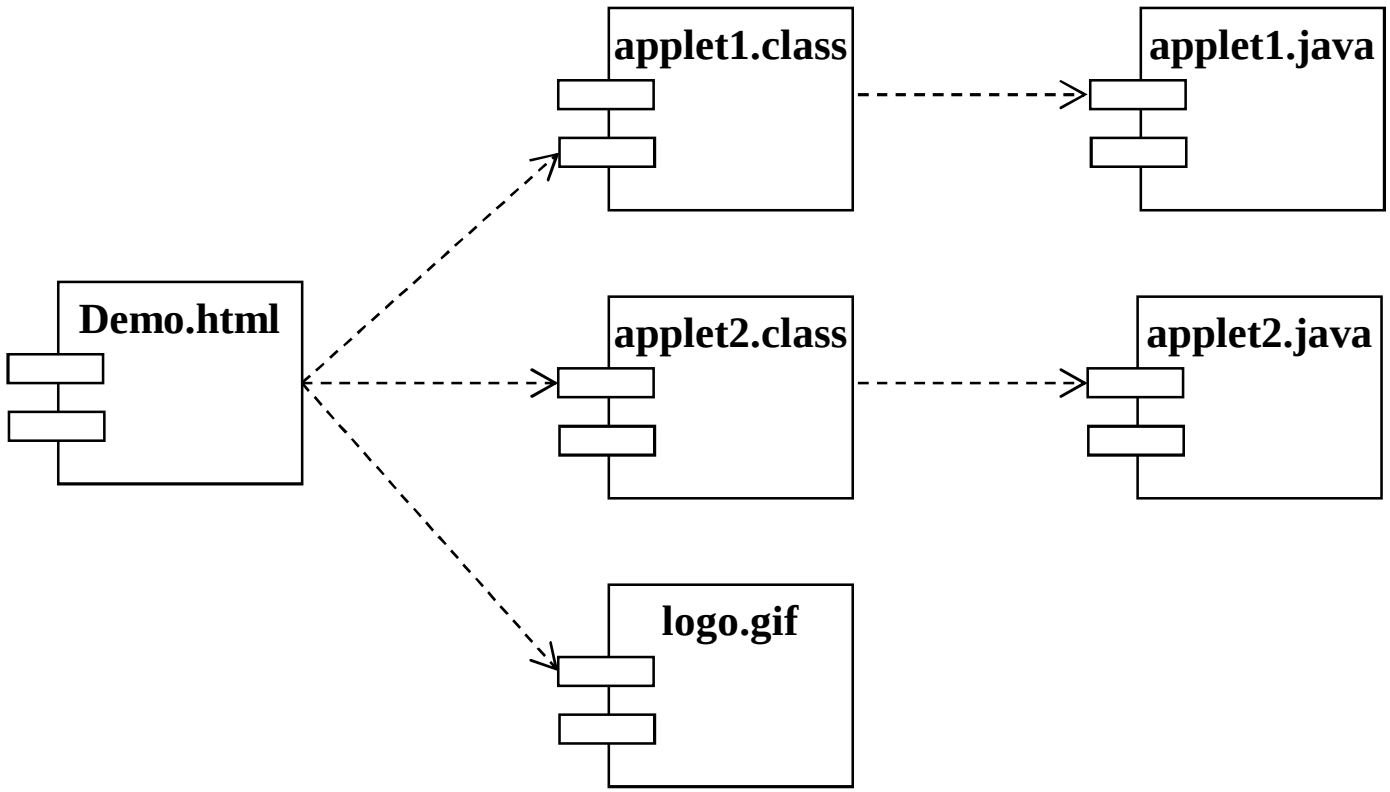
- Menunjukkan arsitektur fisik dan komponen perangkat lunak sistem. *For example, network nodes*



UML Diagrams

Component Diagram

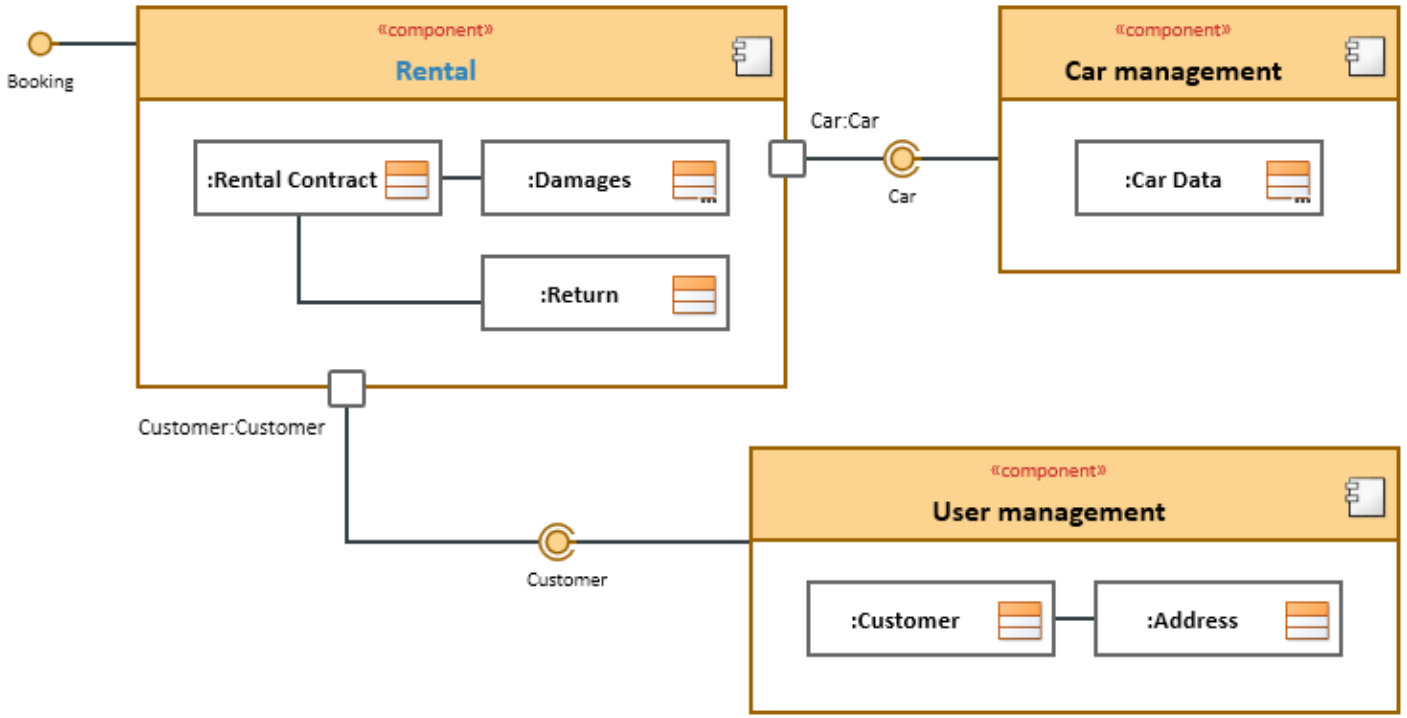
- Hubungan fisik di antara komponen perangkat lunak



UML Diagrams

Composite Structure

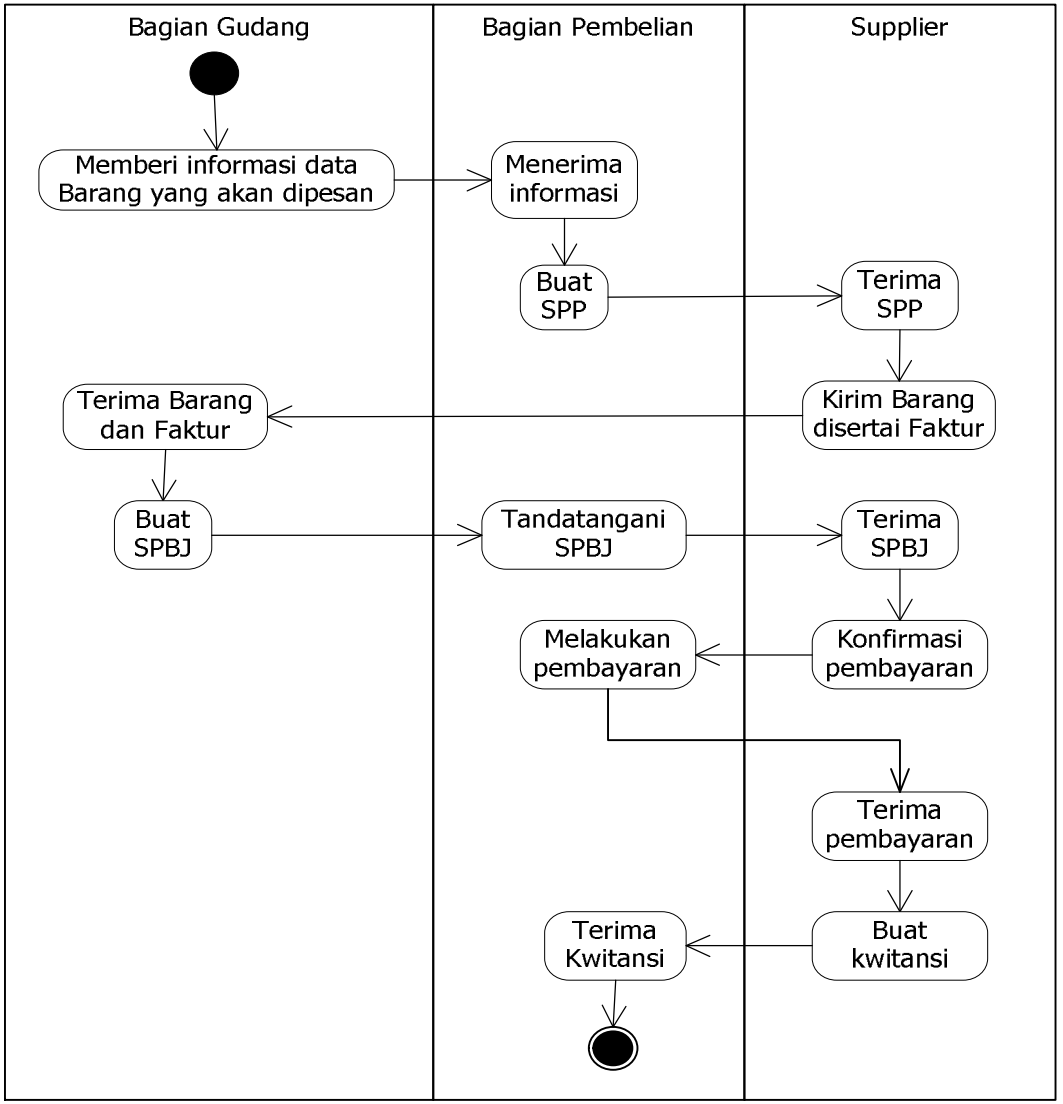
- Menggambarkan struktur internal dari kelas yang kompleks
- Baru di UML 2.0



UML Diagrams

Activity Diagram

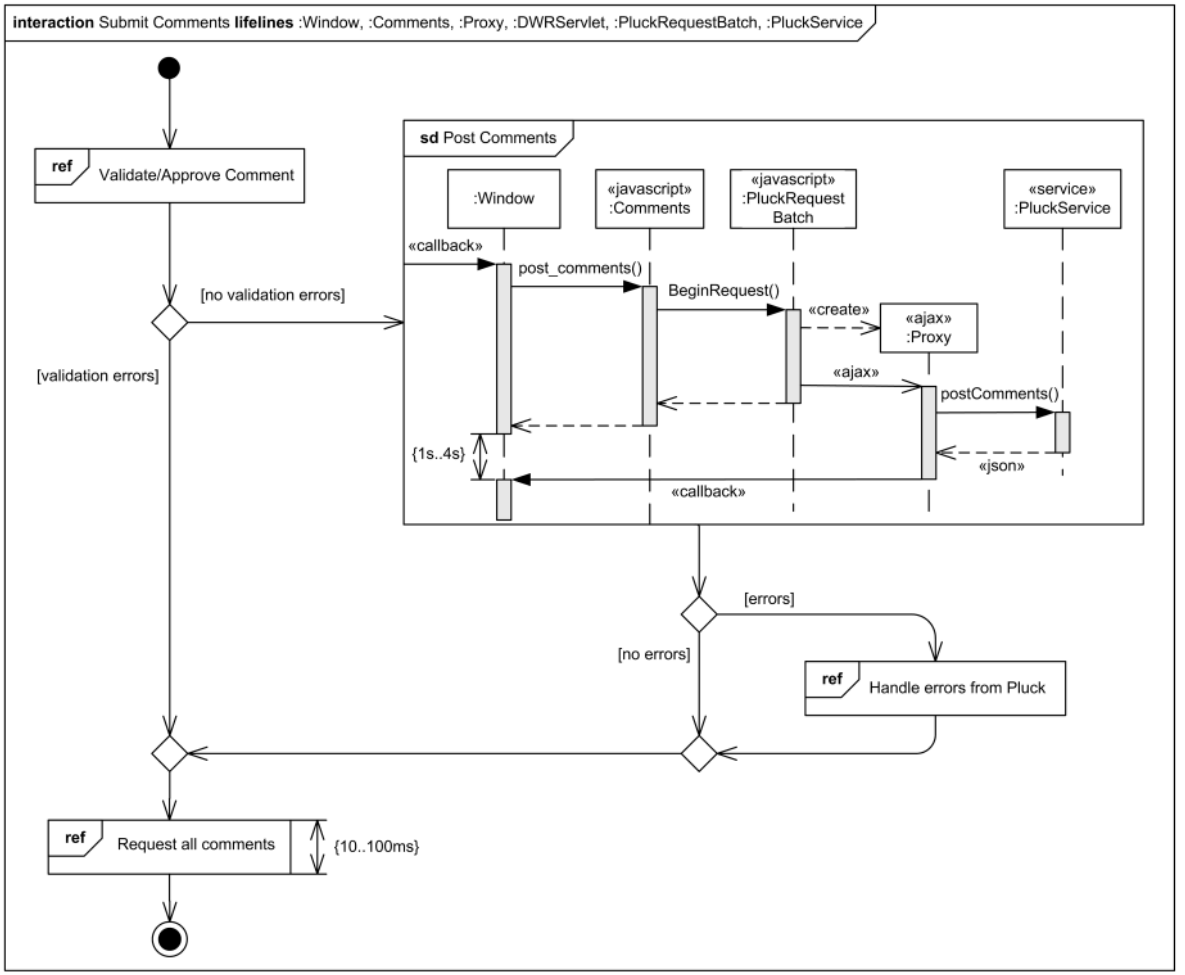
- Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses



UML Diagrams

Interaction Overview Diagram

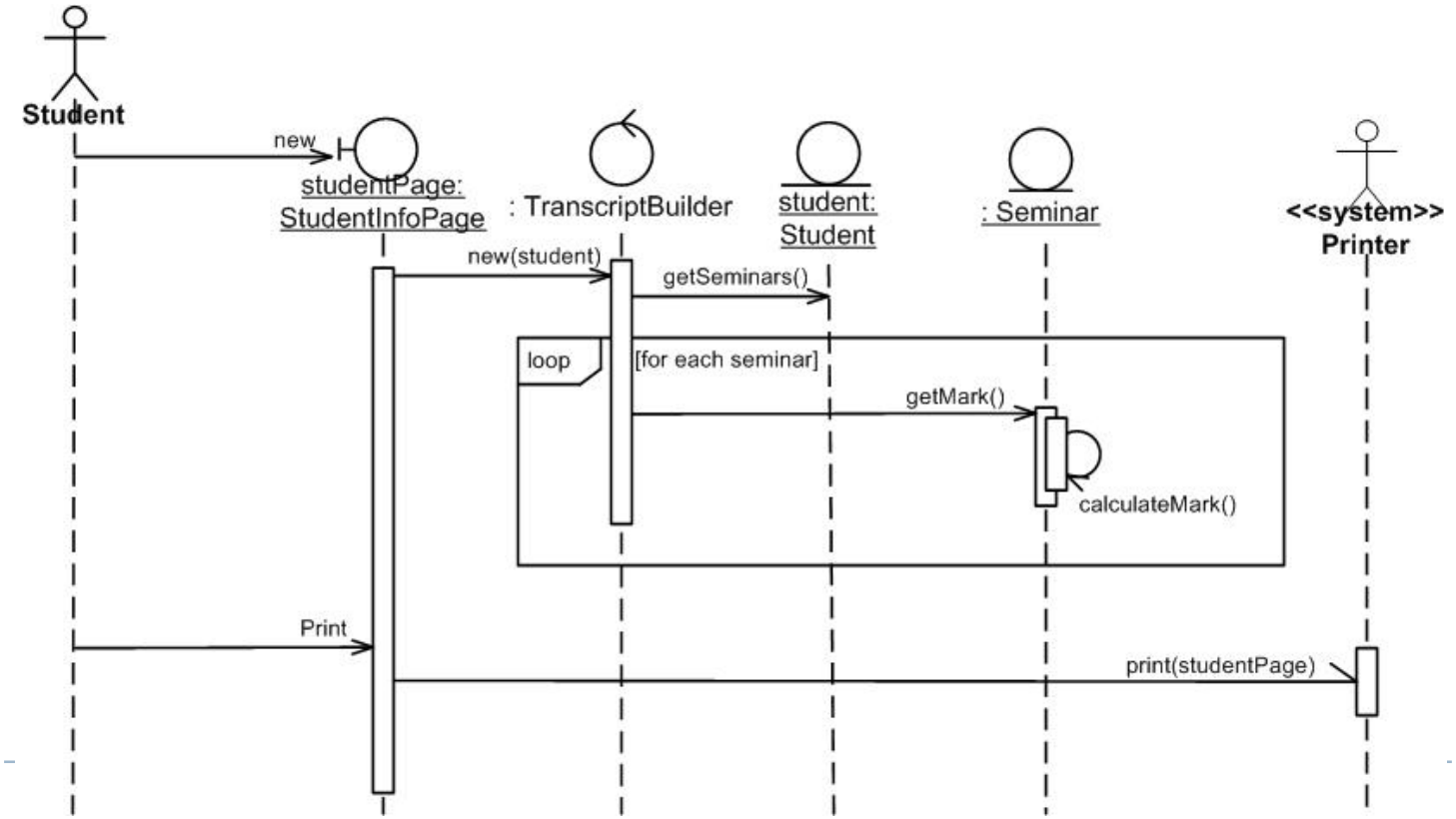
- Campuran dari *activity* dan *sequence diagram*
- Baru di UML 2.0



UML Diagrams

Sequence Diagram

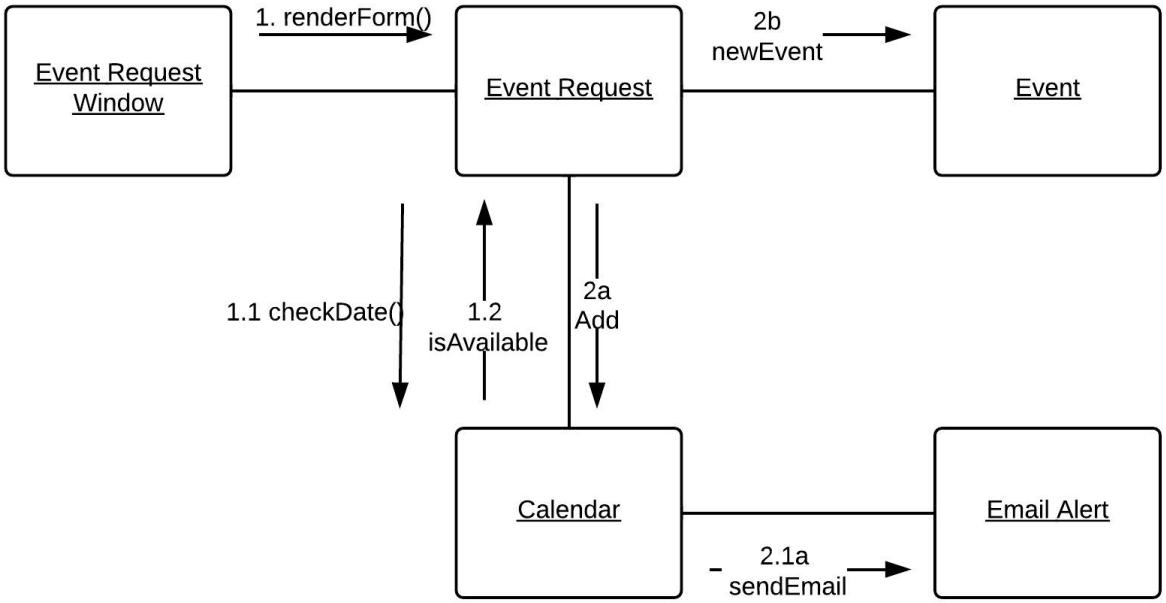
- Menggambarkan interaksi secara berurutan berdasarkan waktu interaksi



UML Diagrams

Comunication Diagram

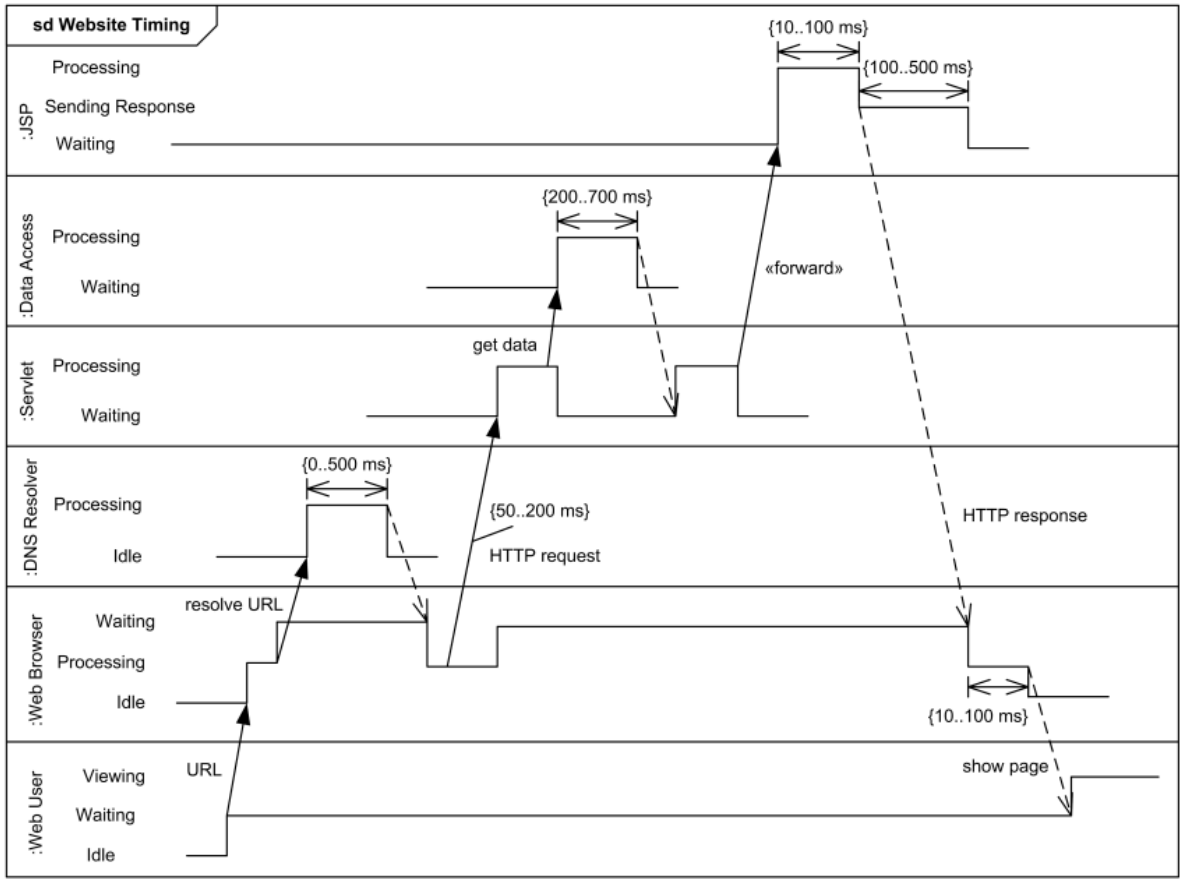
- Menggambarkan komunikasi antara sekumpulan objek
- Menekankan pada jalur
- Pada UML 1.x disebut Diagram Kolaborasi



UML Diagrams

Timming Diagram

- Interaksi antar objek yang menekankan pada waktu (*timming*)
- Baru di UML 2.0

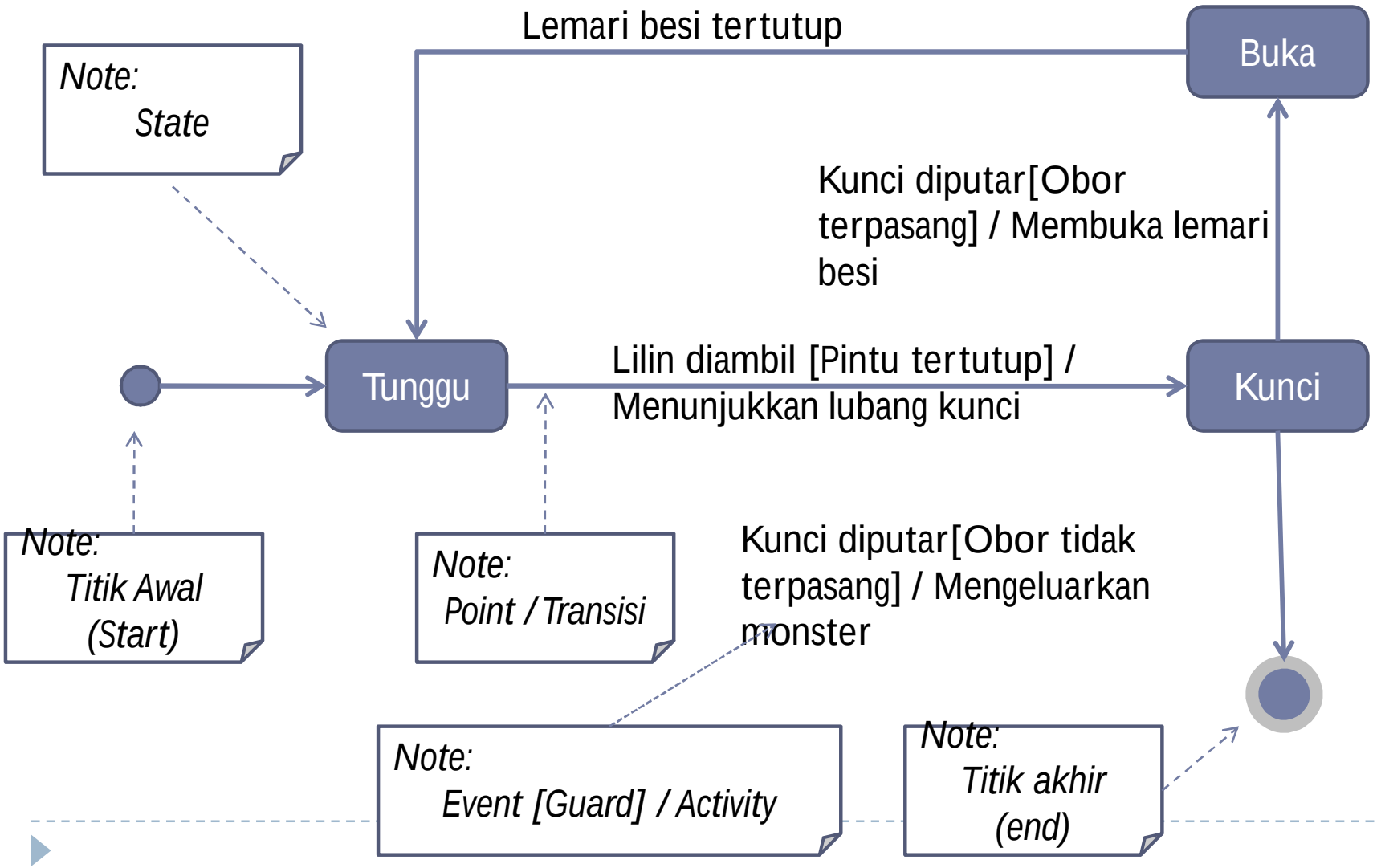


UML Diagram

State Machine Diagram

- Memeriksa perilaku dari suatu kelas
- Menunjukkan model keadaan-keadaan yang berbeda dan transisi keadaan dari suatu objek
- di UML 1.x disebut *Statechart Diagram*

Example State Machine Diagram

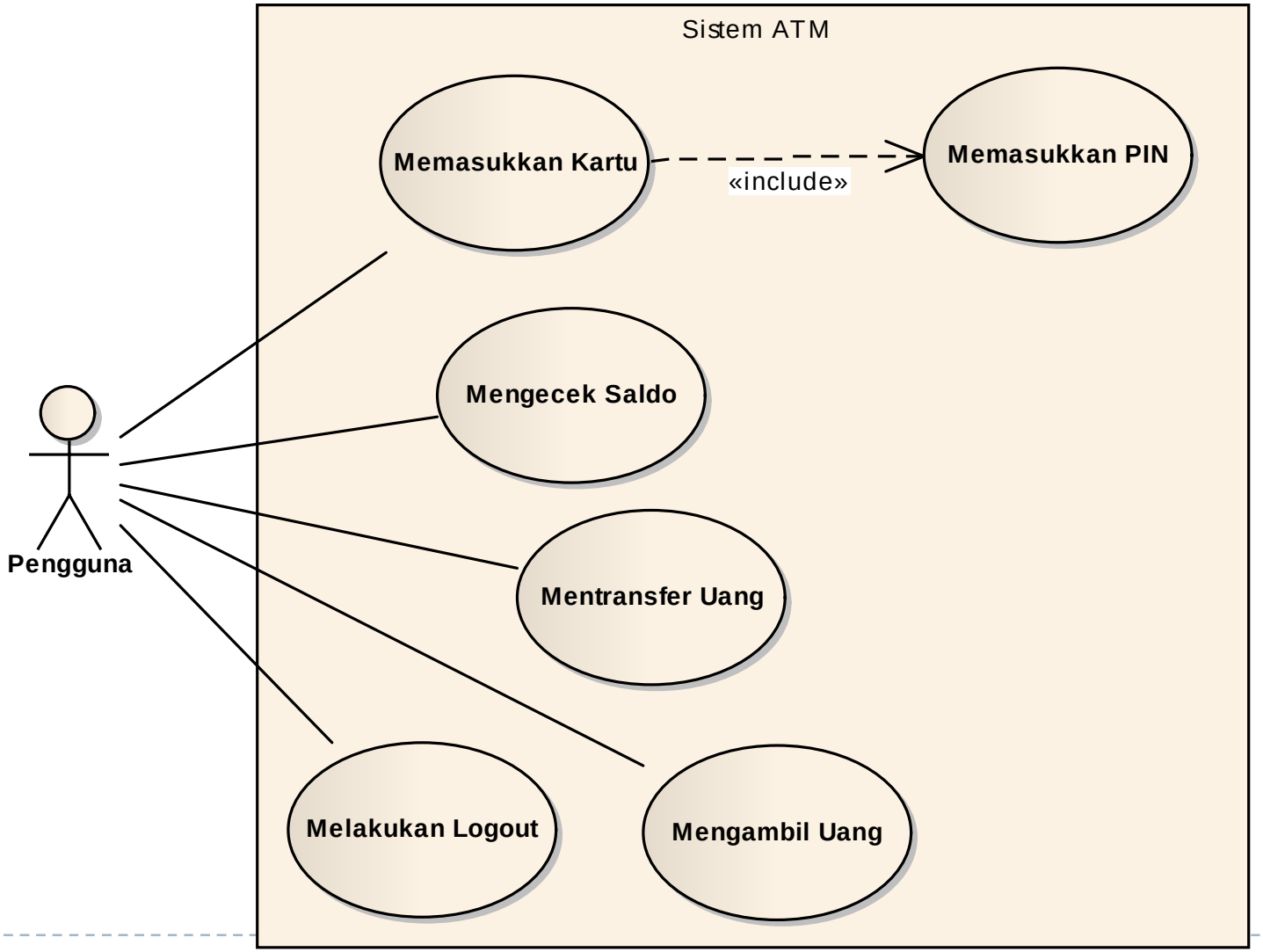


UML Diagrams

Use Case Diagram

- Menunjukkan interaksi antara sistem dan lingkungan
- Menggambarkan fungsionalitas yang diharapkan dari sebuah sistem.
- Menekankan “apa” yang diperbuat sistem, dan bukan “bagaimana”.
- Menggambarkan kebutuhan sistem dari sudut pandang pengguna (*user*)

Exampe Use Case Diagram



UML Tools

- Rational Rose
- Visual Paradigm
- Enterprise Architect
- Microsoft Visio
- Star UML
- Netbeans UML Plugin

Thanks

&

See You

Next Chapter ……