

PENGANTAR UML

Analysis Design Paradigm and Diagrams

	Paradigm	Diagrams
1	Data-oriented	DFD
2	Process-oriented	Flowchart
3	Object-oriented (data + process)	UML

Sejarah UML



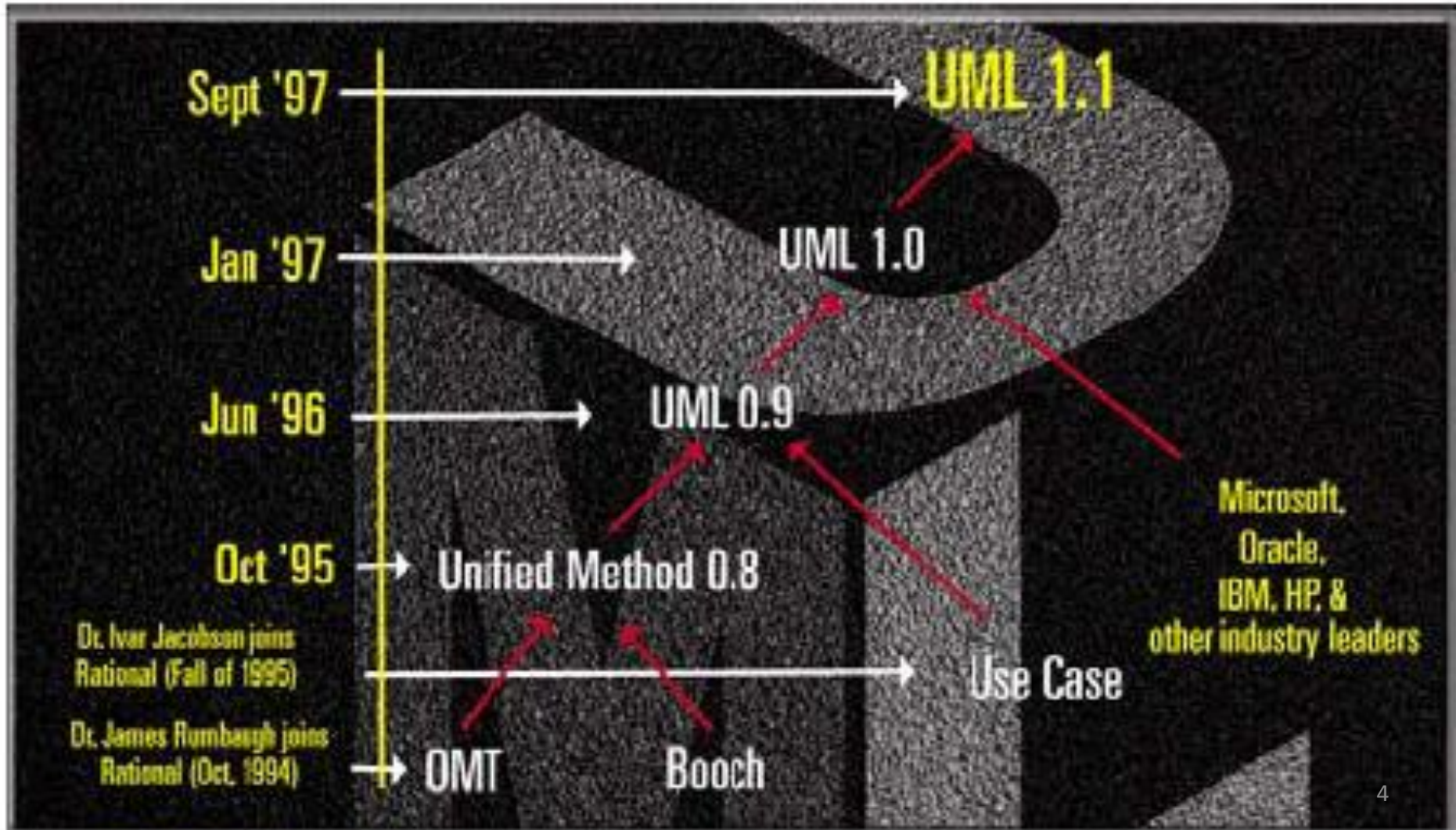
Booch, Jacobson, Rumbaugh

- In the 90s many people creating OO diagramming languages
- Three different ones created by Grady Booch, Ivar Jacobson, James Rumbaugh
- Joined forces with Rational (company) to create Unified Modeling Language (UML)

Sejarah UML

2011 → UML 2.4

2003 → UML 2.0



What is the UML?

- UML: **Unified Modeling Language**
- UML can be used for **modeling all processes in the development life cycle** and across different implementation technologies (technology and language independent)
- UML is the **standard language** for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system
- UML is a **communication tool** – for the team, and other stakeholders

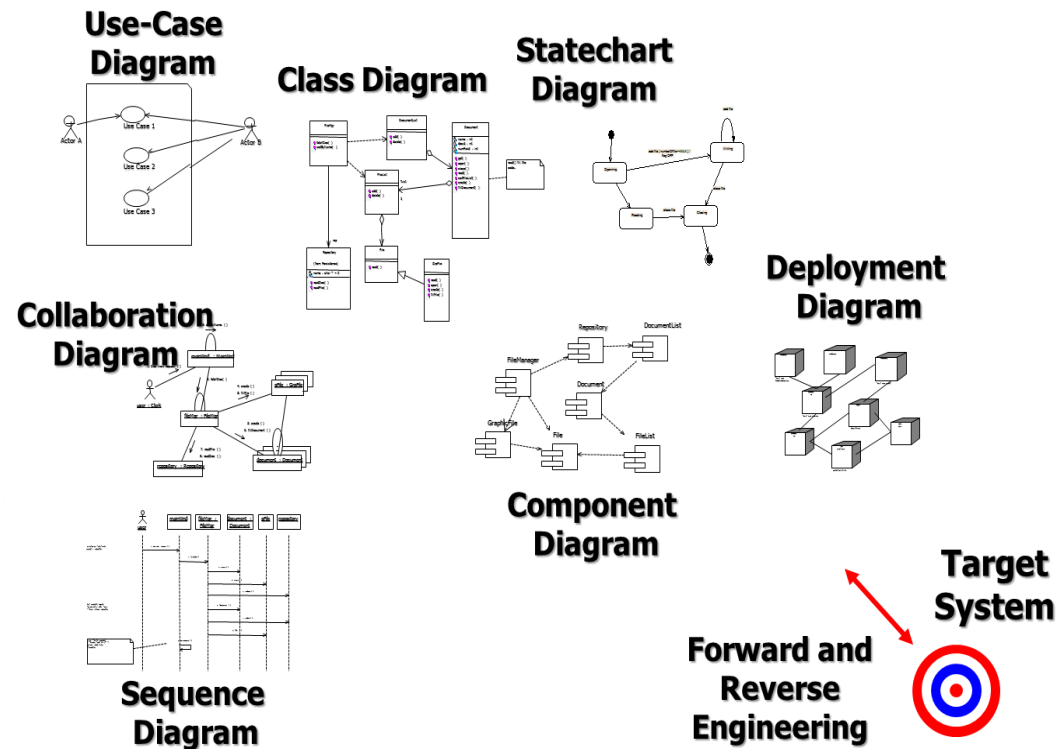
UML Tools

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

UML 2.0 Diagrams

UML version 2.0 has 14 diagrams in 2 major groups:

1. **Structure** Diagrams
2. **Behavior** Diagrams



UML 2.0 Diagram

Diagram Name	Used to	Primary Phase
Structure Diagrams		
Class	Illustrate the relationships between classes modeled in the system.	Analysis, Design
Object	Illustrate the relationships between objects modeled in the system. Used when actual instances of the classes will better communicate the model.	Analysis, Design
Package	Group other UML elements together to form higher level constructs.	Analysis, Design, Implementation
Deployment	Show the physical architecture of the system. Can also be used to show software components being deployed onto the physical architecture.	Physical Design, Implementation
Component	illustrate the physical relationships among the software components.	Physical Design, Implementation
Composite Structure	Illustrate the internal structure of a class, i.e., the relationships among the parts of a class.	Analysis, Design
Behavioral Diagrams		
Activity	Illustrate business workflows independent of classes, the flow of activities in a use case, or detailed design of a method.	Analysis, Design
Sequence	Model the behavior of objects within a use case. Focuses on the time-based ordering of an activity.	Analysis, Design
Communication	Model the behavior of objects within a use case. Focuses on the communication among a set of collaborating objects of an activity.	Analysis, Design
Interaction Overview	Illustrate an overview of the flow of control of a process.	Analysis, Design
Timing	Illustrate the interaction that takes place among a set of objects and the state changes in which they go through along a time axis.	Analysis, Design
Behavioral State Machine	Examine the behavior of one class.	Analysis, Design
Protocol State Machine	Illustrates the dependencies among the different interfaces of a class.	Analysis, Design
Use-Case	Capture business requirements for the system and to illustrate the interaction between the system and its environment.	Analysis

UML Structure Diagrams

Represent the **data and static relationships** in an information system

1. **Class** Diagram
2. **Object** Diagram
3. **Package** Diagram
4. **Deployment** Diagram
5. **Component** Diagram
6. **Composite** Structure Diagram

Structure Diagrams

1. Class Diagrams

- Common vocabulary used by analyst and users
- Represent things (employee, paycheck,...)
- Shows the relationships between classes

2. Object Diagrams

- Similar to class diagrams
- Instantiation of a class diagram
- Relationships between objects

3. Package Diagrams

- Group UML elements together to form higher level constructs

Structure Diagrams

4. Deployment Diagrams

- Shows the **physical architecture** and software components of system
- For example, network nodes

5. Component Diagrams

- **Physical relationships** among software components
- Example – Client/Server (Which machines run which software)

6. Composite Structure

- Illustrates internal structure of a complex class

UML Behavior Diagrams

Depict the dynamic relationships among the instances or objects that represent the business information system

1. Activity Diagram
2. Sequence Diagram
3. Communication Diagram
4. Interaction Diagram
5. Timing Diagram
6. Behavior State Machine
7. Protocol State Machine
8. Use Case Diagrams

Behavior Diagrams

1. Activity Diagrams

- Model processes in an information system
- Example: Business workflows, business logic

2. Interaction Diagrams

- Shows interaction among objects

3. Sequence Diagrams

- Time-based ordering of the interaction

4. Communication Diagrams

- Communication among a set of collaborating objects of an activity

Behavior Diagrams

5. Interaction Diagrams

- Overview of flow of control of a process

6. Timing Diagrams

- Show how an object changes over time

7. State Machines

- Examines behavior of one class
- Models the different states and state transitions an object can experience

8. Use-Case Diagrams

- Shows **interaction between the system and environment**
- Captures business requirements

UML Problems

1. UML is modeling notation, it is **not a development process** or a methodology
 - UML driven development process?
2. UML is **too complex, difficult to understand quickly**
 - Should we use all UML diagrams?

UML Process (EA Sparx)

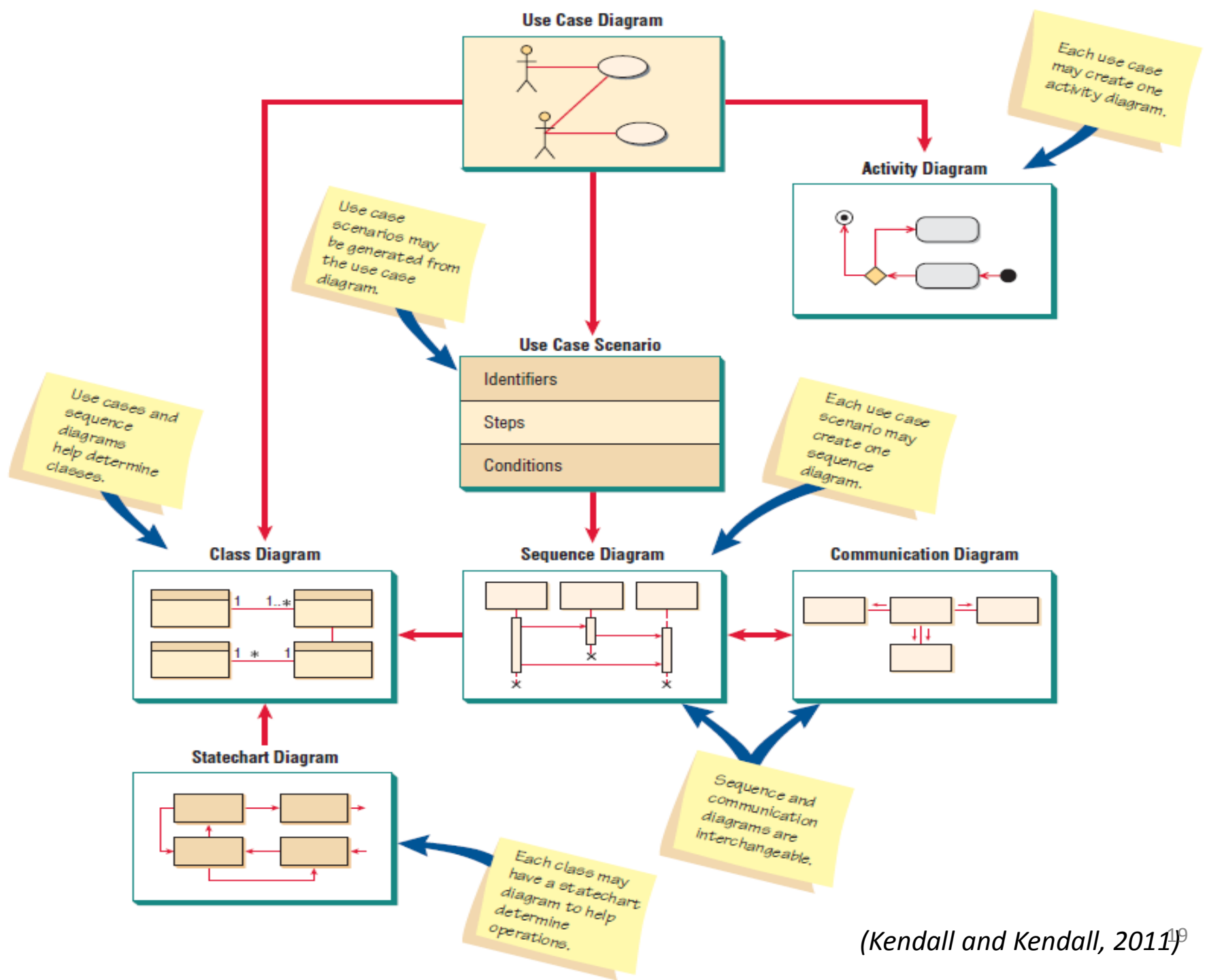
1. Display the boundary of a system and its major functions using **use cases and actors**
2. Model the organization's business process with **activity diagram**
3. Illustrate use case realizations with **sequence diagrams**
4. Represent a static structure of a system using **class diagrams**
5. Reveal the physical implementation architecture with **deployment diagrams**

UML Process (EA Sparx)

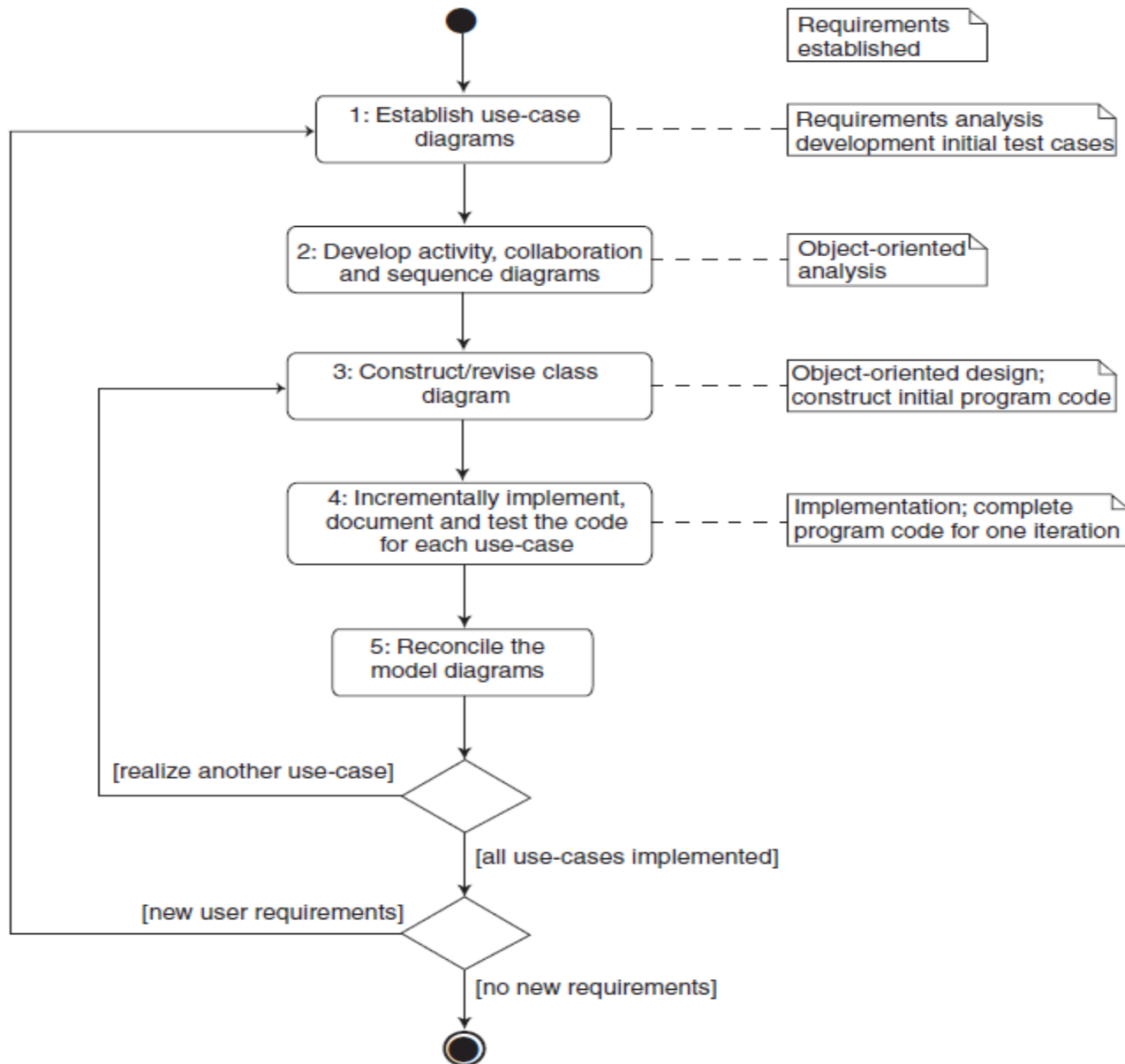
1. Use Cases Diagram
2. Activity Diagram
3. Sequence Diagram
4. Class Diagram
5. Deployment Diagrams

UML Process (Kendal, 2011)

1. A **use case diagram**, describing how the system is used. Analysts start with a use case diagram
2. An **activity diagram**, illustrating the overall flow of activities. Each use case may create one activity diagram
3. **Sequence diagrams**, showing the sequence of activities and class relationships. Each use case may create one or more sequence diagrams
4. **Class diagrams**, showing the classes and relationships. Sequence diagrams are used to determine classes
5. **Statechart diagrams**, showing the state transitions. Each class may create a statechart diagram, which is useful for determining class methods



UML Process (Barclay, 2004)



Bangunan Dasar UML

1. Sesuatu (Things)
2. Relasi (Relationship)
3. Diagram

Things

Ada 4 Macam Things dalam UML :

- a. Structural Things
- b. Behavioral Things
- c. Grouping Things
- d. Annotational Things

A. Structural Things

- Merupakan Bagian yang bersifat statis dalam model UML
- Dapat berupa elemen-elemen yang bersifat fisik maupun konseptual
- Ada 7 macam structural things, yaitu Kelas, Antarmuka, Kolaborasi, Use Case, Kelas Aktif, Komponen dan Simpul

7 macam structural things (1)

Kelas

- Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
- Digambarkan dengan empat-persegi-panjang yang memuat nama, atribut, serta operasi yang dimilikinya

7 macam structural things (2)

Antarmuka (Interfaces)

- Kumpulan dari operasi-operasi yang menspesifikasi layanan (service) suatu kelas atau komponen/objek
- Mendeskripsikan perilaku yang tampak dari luar dari suatu elemen
- Jarang berdiri sendiri.
- Biasanya dilampirkan pada kelas atau komponen yang merealisasikan antarmuka
- secara grafis digambarkan dengan lingkaran kecil dengan namanya didahului dengan garis tegak (|)

7 macam structural things (3)

Kolaborasi (Collaboration)

- Mendefinisikan interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dari elemen-elemennya (sinergi)
- Merepresentasikan pola implementasi yang memperbaiki sistem
- secara grafis digambarkan dengan elipsi bergaris putus-putus yang memuat nama kolaborasi itu.

7 macam structural things (4)

Use Case

- Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
- Digunakan untuk menstrukturkan perilaku pada suatu model
- Digambarkan dengan elips tegas yang berisi namanya

7 macam structural things (5)

Kelas Aktif (Active Class)

- Kelas dimana Objek-objek yang dimilikinya memiliki satu atau lebih proses dan lebih jauh menginisialisasi suatu objek kendali.
- Merupakan kelas biasa namun objek-objek yang dimilikinya menampilkan elemen-elemen yang memiliki perilaku konkuren.
- secara grafis digambarkan seperti kelas biasa tetapi dengan batas yang *lebih tebal*, yang memuat nama, atribut, serta operasi yang dimilikinya.

7 macam structural things (6)

Komponen (Component)

- Bagian fisik dan bagian yang dapat digantikan pada suatu sistem.
- Secara grafis digambarkan dengan empat-persegi-panjang seperti kelas tetapi ditambahi tab.

7 macam structural things (7)

Simpul (Node)

- Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.
- Kumpulan komponen mungkin hadir dalam simpul dan mungkin juga berpindah-pindah dari suatu simpul ke simpul yang lain.
- secara grafis digambarkan sebagai kubus yang berisi namanya.

B. Behavioral Things

- Merupakan bagian yang dinamis pada model UML
- Mencerminkan perilaku sepanjang ruang dan waktu.
- Ada 2 macam behavioral things :
 1. Interaksi
 2. State

Behavioral Things

Interaksi

- Suatu perilaku yang mencakup himpunan pesan-pesan yang diperlukan untuk menyelesaikan suatu fungsi tertentu
- Terdiri dari pesan-pesan, urutan aksi (perilaku yang dihasilkan oleh sebuah pesan), link (hubungan antar objek-objek)
- Secara grafis, pesan digambarkan dengan tanda panah tegas yang sering memuat nama operasinya

Behavioral Things

State

- Perilaku yang menspesifikasi unsur kedudukan suatu objek atau interaksi-interaksi sepanjang waktu dalam menanggapi event-event yang terjadi.
- Penggambaran suatu state memuat beberapa unsur yaitu state itu sendiri, transisi (perubahan dari suatu state ke state lainnya), event (suatu keadaan yang memicu sebuah transisi, serta aktivitas (tanggapan terhadap transisi)
- Digambarkan sebagai empat-persegi-panjang yang sudut-sudutnya melengkung, yang memuat namanya (serta substate didalamnya, jika ada)

C. Grouping Things

- Bagian pengorganisasi dalam UML
- Dalam penggambaran model UML yang rumit kadang diperlukan penggambaran paket yang menyederhanakan model
- Paket berguna bagi mengelompokkan sesuatu, misalnya model-model serta subsistem-subsistem.

D. Annotational Things

- Bagian yang memperjelas model UML
- Dapat berupa komentar yang memperjelas fungsi serta ciri-ciri tiap elemen dalam model UML

RELATIONSHIP

- Hubungan-hubungan yang terjadi antarelelemen dalam UML
- Ada 4 macam relationship dalam UML, yaitu Dependency, Asosiasi, Generalisasi, Realisasi

Dependency (Kebergantungan)

- Hubungan dimana perubahan yang terjadi pada suatu elemen independen (mandiri) akan mempengaruhi elemen yang bergantung padanya (elemen yang tidak mandiri – dependen).
- Secara grafis digambarkan dengan tanda panah putus-putus.

Asosiasi

- Menghubungkan antara objek satu dengan objek yang lainnya; bagaimana hubungan suatu objek dengan objek lainnya.
- Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.
- Secara grafis digambarkan dengan garis tegas tanpa tanda panah.

Generalisasi

- Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya (objek induk – ancestor).
- Arah dari atas ke bawah (dari objek induk ke objek anak disebut spesialisasi)
- Arah dari bawah ke atas disebut generalisasi
- Secara grafis digambarkan sebagai garis yang ujungnya berkepala panah (atau bentuk segitiga) yang kosong, yang mengarah ke objek induk.

Realisasi

- Operasi yang benar-benar dilakukan oleh suatu objek
- Secara grafis digambarkan dengan tanda panah bergaris putus-putus dengan kepala panah kosong.

Diagram

Ada 9 jenis diagram, yaitu :

1. Diagram Kelas
2. Diagram Objek
3. Use Case Diagram
4. Sequence Diagram
5. Collaboration Diagram
6. Statechart Diagram
7. Activity Diagram
8. Component Diagram
9. Deployment Diagram

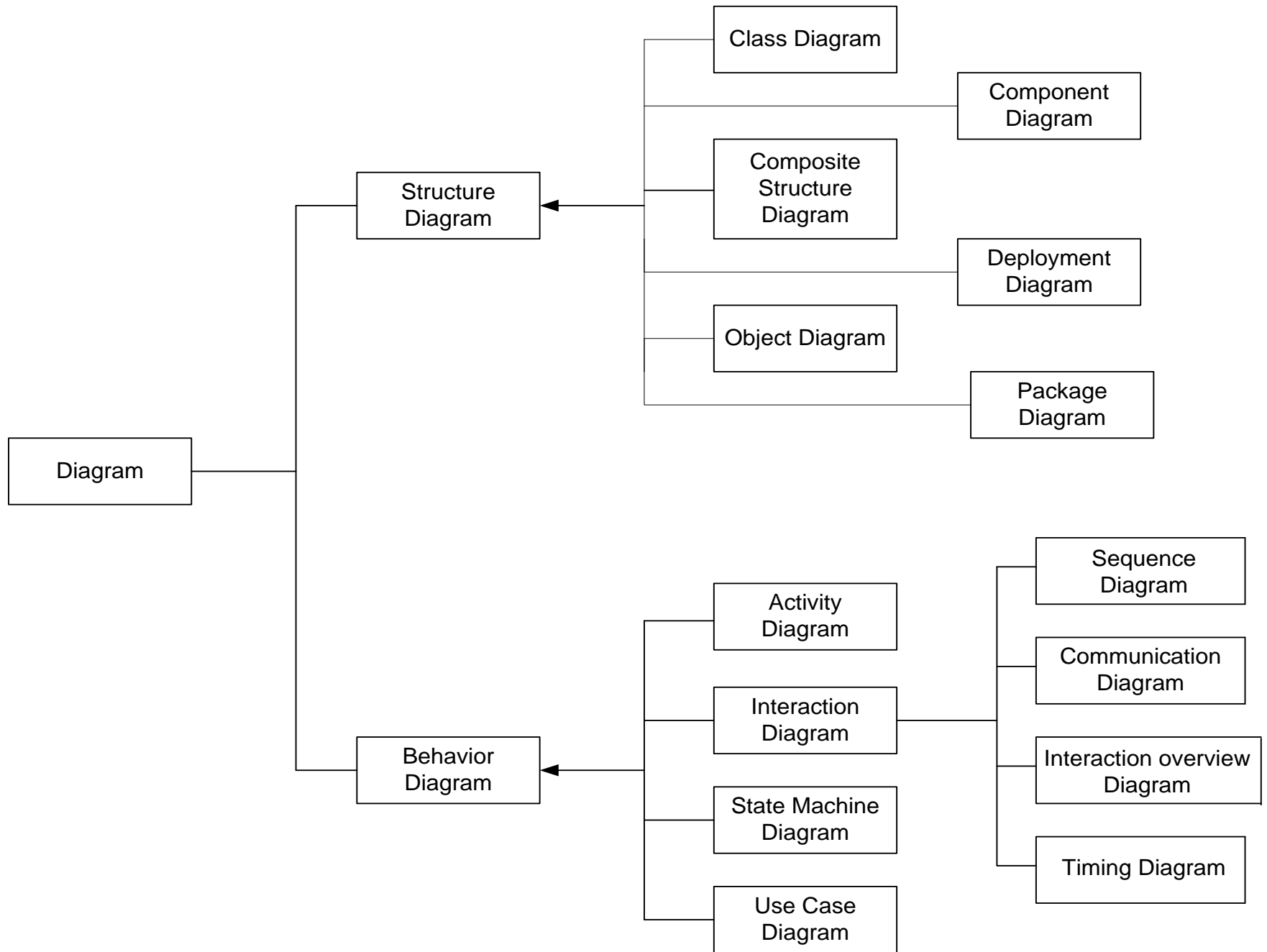


Diagram Kelas

- Bersifat Statis
- Memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi

Diagram Objek

- Bersifat statis
- Menunjukkan objek-objek serta relasi-relasi antarobjek.
- Menunjukkan instantiasi statis dari segala sesuatu yang dijumpai pada diagram kelas

Use-Case Diagram

- Bersifat statis
- Memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas).
- Berfungsi mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan dan diharapkan pengguna.

Sequence Diagram

- Bersifat dinamis
- Diagram yang menekankan pada pengiriman pesan dalam suatu waktu tertentu

Collaboration Diagram

- Bersifat dinamis
- Diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan

Statechart Diagram

- Bersifat dinamis
- Menunjukkan state-state pada sistem; memuat state, transisi, event, serta aktivitas.
- Menunjukkan sifat dinamis dari antarmuka, kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif

Activity Diagram

- Bersifat dinamis
- Menunjukkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem.
- Memodelkan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

Component Diagram

- Bersifat statis
- Menunjukkan organisasi serta kebergantungan pada komponen-komponen yang telah ada sebelumnya
- Berhubungan dengan diagram kelas.

Deployment Diagram

- Bersifat statis
- Menunjukkan konfigurasi saat aplikasi dijalankan (saat runtime).
- Memuat node beserta komponen-komponen yang ada didalamnya
- Berhubungan dengan diagram komponen dimana deployment diagram memuat satu atau lebih komponen-komponen

Langkah-langkah Penggunaan UML (1)

- Buatlah daftar business process dari level tertinggi untuk mendefinisikan aktivitas dan proses yang mungkin muncul.
- Petakan use case untuk tiap business process untuk mendefinisikan dengan tepat fungsionalitas yang harus disediakan oleh sistem. Kemudian perhalus use case diagram dan lengkapi dengan requirement, constraints dan catatan-catatan lain.
- Buatlah deployment diagram secara kasar untuk mendefinisikan arsitektur fisik sistem.
- Definisikan requirement lain (non-fungsional, security dan sebagainya) yang juga harus disediakan oleh sistem.
- Berdasarkan use case diagram, mulailah membuat activity diagram.

Langkah-langkah Penggunaan UML (2)

- Definisikan objek-objek level atas (package atau domain) dan buatlah sequence dan/atau collaboration diagram untuk tiap alir pekerjaan. Jika sebuah use case memiliki kemungkinan alir normal dan error, buatlah satu diagram untuk masing-masing alir.
- Buarlah rancangan user interface model yang menyediakan antarmuka bagi pengguna untuk menjalankan skenario use case.
- Berdasarkan model-model yang sudah ada, buatlah class diagram. Setiap package atau domain dipecah menjadi hirarki class lengkap dengan atribut dan metodenya. Akan lebih baik jika untuk setiap class dibuat unit test untuk menguji fungsionalitas class dan interaksi dengan class lain.
- Setelah class diagram dibuat, kita dapat melihat kemungkinan pengelompokan class menjadi komponen-komponen. Karena itu buatlah component diagram pada tahap ini. Juga, definisikan tes integrasi untuk setiap komponen meyakinkan ia berinteraksi dengan baik.

Langkah-langkah Penggunaan UML

(3)

- Perhalus deployment diagram yang sudah dibuat. Detilkan kemampuan dan requirement piranti lunak, sistem operasi, jaringan, dan sebagainya. Petakan komponen ke dalam node.
- Mulailah membangun sistem. Ada dua pendekatan yang dapat digunakan
 - Pendekatan use case, dengan meng-assign setiap use case kepada tim pengembang tertentu untuk mengembangkan unit code yang lengkap dengan tes.
 - Pendekatan komponen, yaitu meng-assign setiap komponen kepada tim pengembang tertentu.
- Lakukan uji modul dan uji integrasi serta perbaiki model berserta codenya. Model harus selalu sesuai dengan code yang aktual
- Piranti lunak siap dirilis.

Latihan

1. Apakah yang dimaksud dengan interaksi dalam UML?
2. Apakah yang dimaksud dengan state dalam UML?
3. Fungsi dari usecase diagram adalah?
4. Apa yang dimaksud dengan statechart diagram?
5. Diagram apakah yang menekankan pada aliran aktivitas?

Latihan

6. Apa yang dimaksud dengan dependency?
7. Gambarkan simbol dependency!
8. Apa yang dimaksud dengan generalisasi dalam UML?
9. Apakah yang dimaksud dengan sequence diagram?
10. Apa yang dimaksud dengan Component diagram?