



**CODELABS**  
BUILD SOCIETY WITH TECHNOLOGY

# REKAYASA PERANGKAT LUNAK II

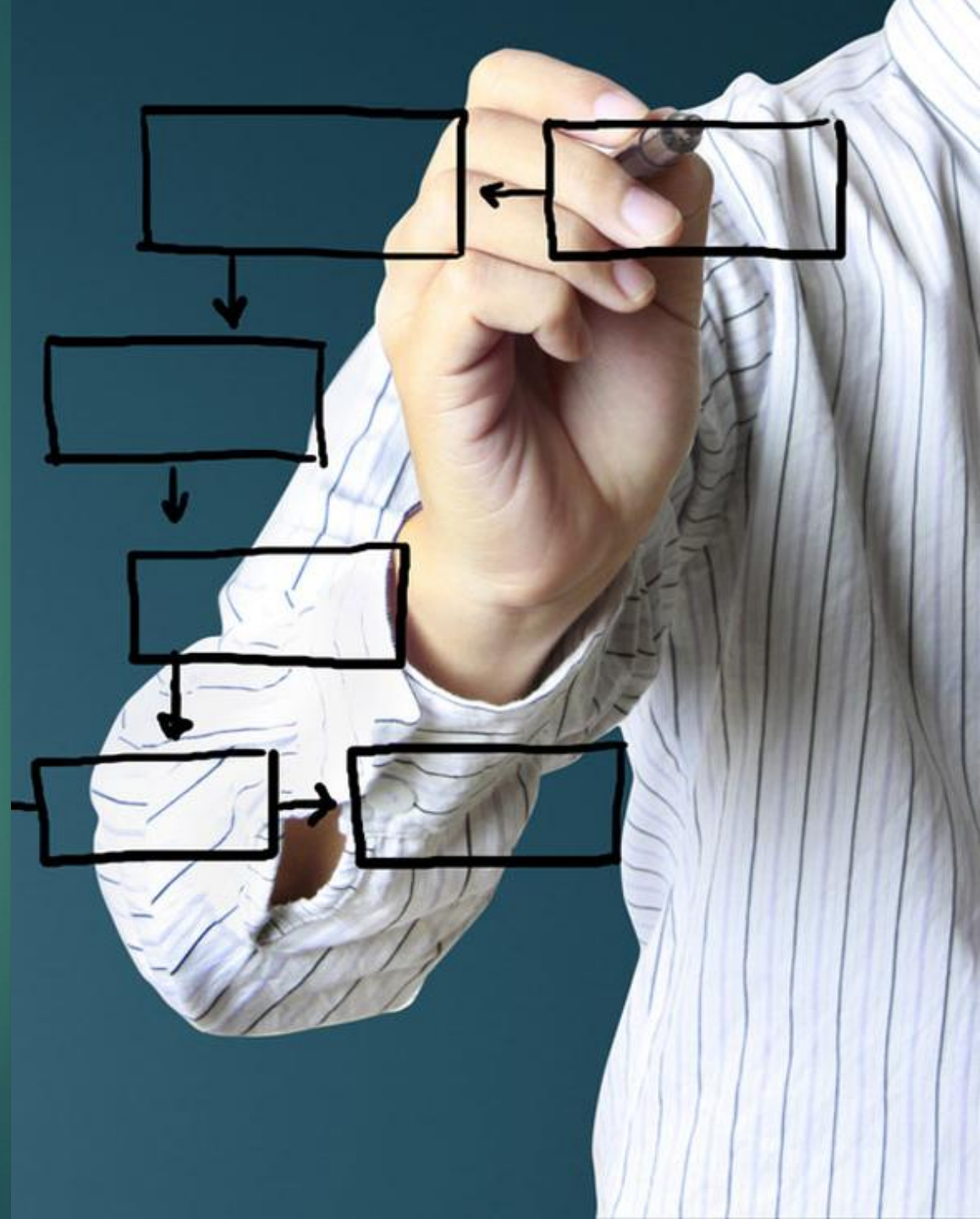
Pendahuluan Rekayasa Perangkat Lunak II

DISUSUN OLEH:

ADAM MUKHARIL BACHTIAR

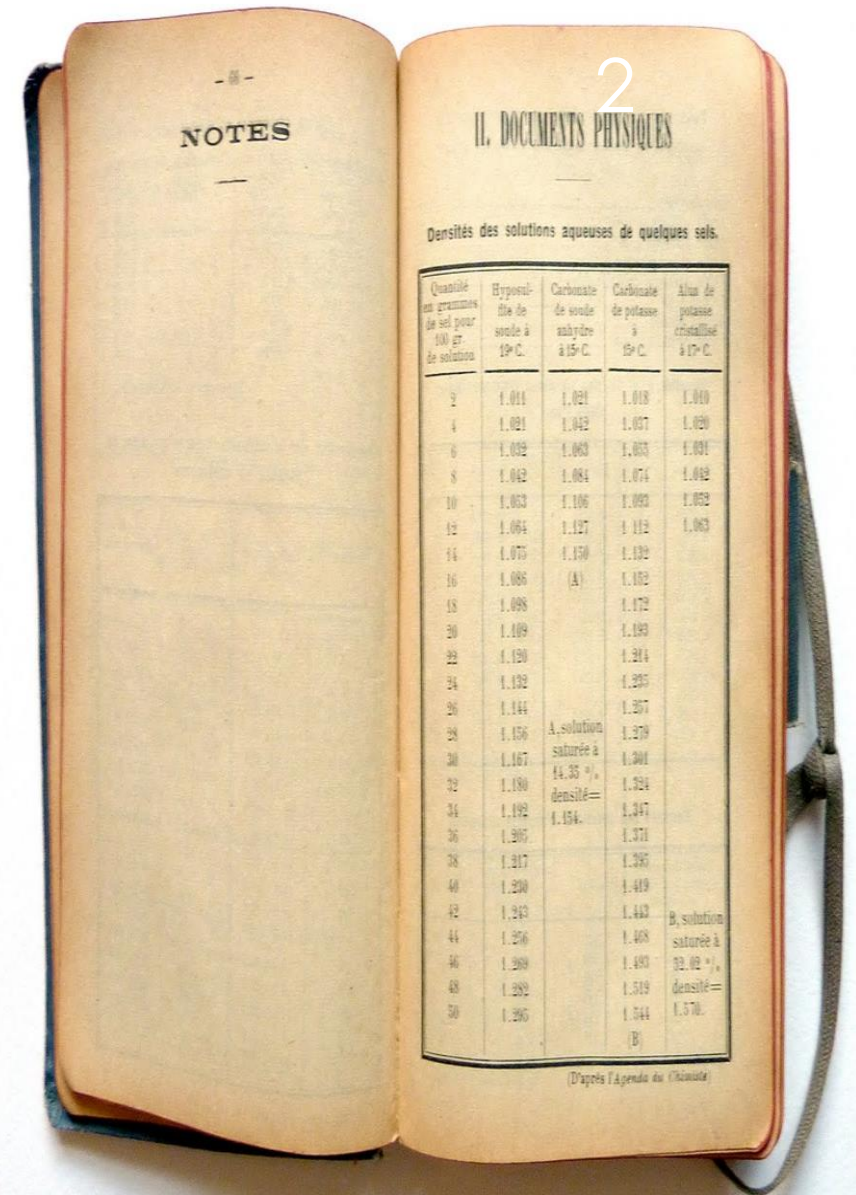
TEKNIK INFORMATIKA UNIKOM

adfbipotter@gmail.com



# AGENDA PERKULIAHAN

- ➔ Deskripsi dan Silabus Mata Kuliah
- ➔ Penilaian dan Referensi
- ➔ Pengantar Rekayasa Perangkat Lunak II



# Deskripsi dan Silabus Mata Kuliah

# Deskripsi Mata Kuliah

4

➡ Sifat : Wajib

➡ Prasyarat : Rekayasa Perangkat Lunak I

➡ Waktu : 3 SKS



# Silabus Mata Kuliah

5

- ➡ Pendahuluan Rekayasa Perangkat Lunak II
- ➡ Pengantar OOAD
- ➡ UML (Per diagram untuk analisis dan desain)

# Penilaian dan Referensi

# Komponen Penilaian

7

Indeks	Nilai
A	$80 \leq NA \leq 100$
B	$68 \leq NA \leq 79$
C	$56 \leq NA \leq 67$
D	$45 \leq NA \leq 55$
E	$0 \leq NA \leq 44$

**30%** TUGAS + **30%** UTS + **40%** UAS

- ➡ Roger S. Pressman, Software Engineering: A Practitioner's Approach, 7th edition.
- ➡ Bernd Bruegge & Allen H. Dutoit, Object-Oriented Software Engineering Using UML, Patterns, and Java
- ➡ Kendall, System Analysis and Design, 8th edition.
- ➡ Slide perkuliahan.



# Pengantar Rekayasa Perangkat Lunak II

# AGENDA PERKULIAHAN

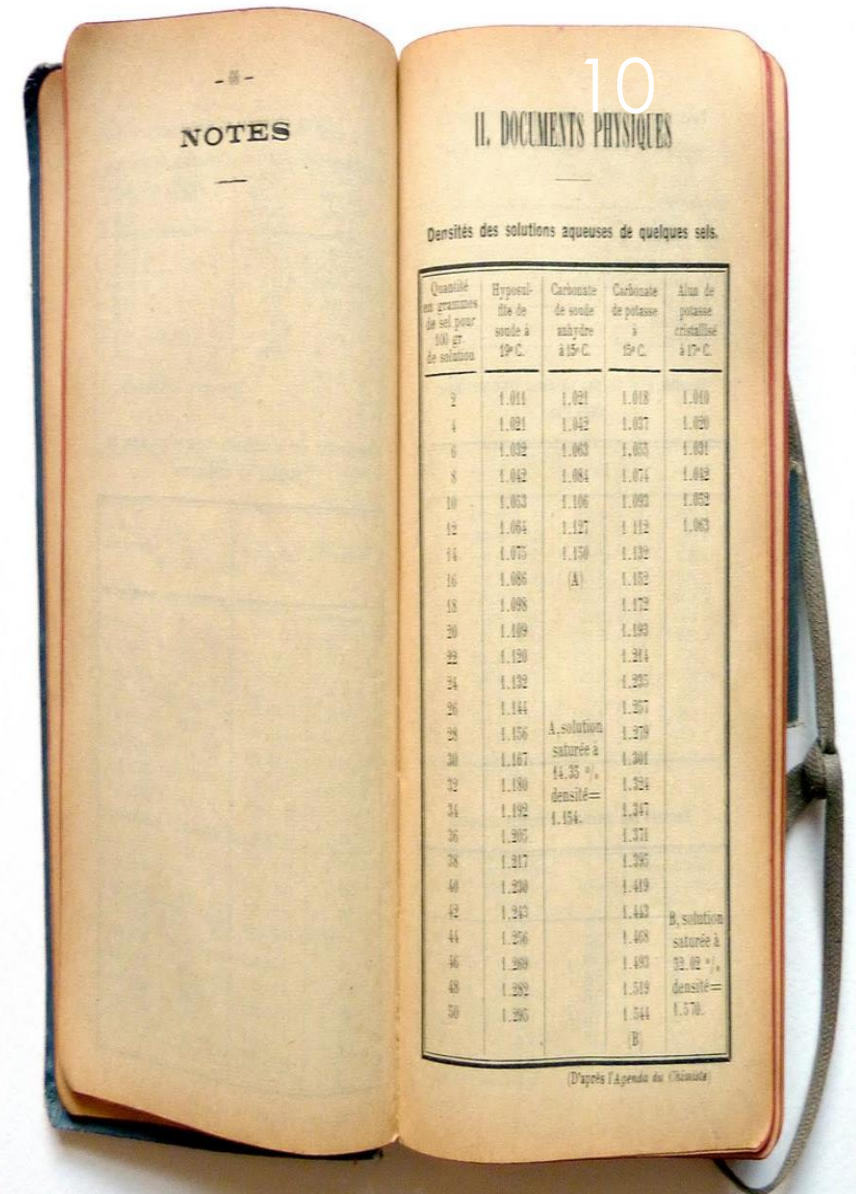
⇒ OOAD

⇒ Objek VS Kelas

⇒ OOAD VS OOP

⇒ Konsep Dasar OOAD

⇒ UML Sebagai tools OOAD



# OOAD

11

- ❶ Singkatan dari **Object Oriented Analysis and Design**
- ❷ Salah satu **pendekatan analisis dan desain** yang bisa digunakan selain analisis terstruktur.
- ❸ OOAD **bukan dipilih berdasarkan bahasa pemrograman** yang digunakan.
- ❹ Pola pikir yang menitik beratkan pada **perekayasaan objek beserta relasinya**.

# Pengertian OOAD

12

- ❶ **Analysis** - Pemahaman, menemukan, dan mendeskripsikan konsep di dalam satu domain masalah.
- ❷ **Design** - Pemahaman dan pendeskripsian solusi dari perangkat lunak yang merepresentasikan konsep analisis dan akan diimplementasikan pada kode program yang dibuat.
- ❸ **OOAD** - Analisis dan Desain menggunakan pendekatan orientasi objek. Sebuah pendekatan pembangunan perangkat lunak yang menekankan pada solusi logika berbasis objek.

# Objek

13





# Definisi Objek

14

- ① Objek **adalah konsepsi atau benda di dunia nyata** yang bisa dibedakan satu dengan yang lainnya.
- ② Objek dapat dibentuk dari **domain permasalahan** yang diambil.
- ③ Objek mempunyai **identitas, properti, dan tingkah laku**.
- ④ Objek merupakan hasil **instansiasi** dari kelas.



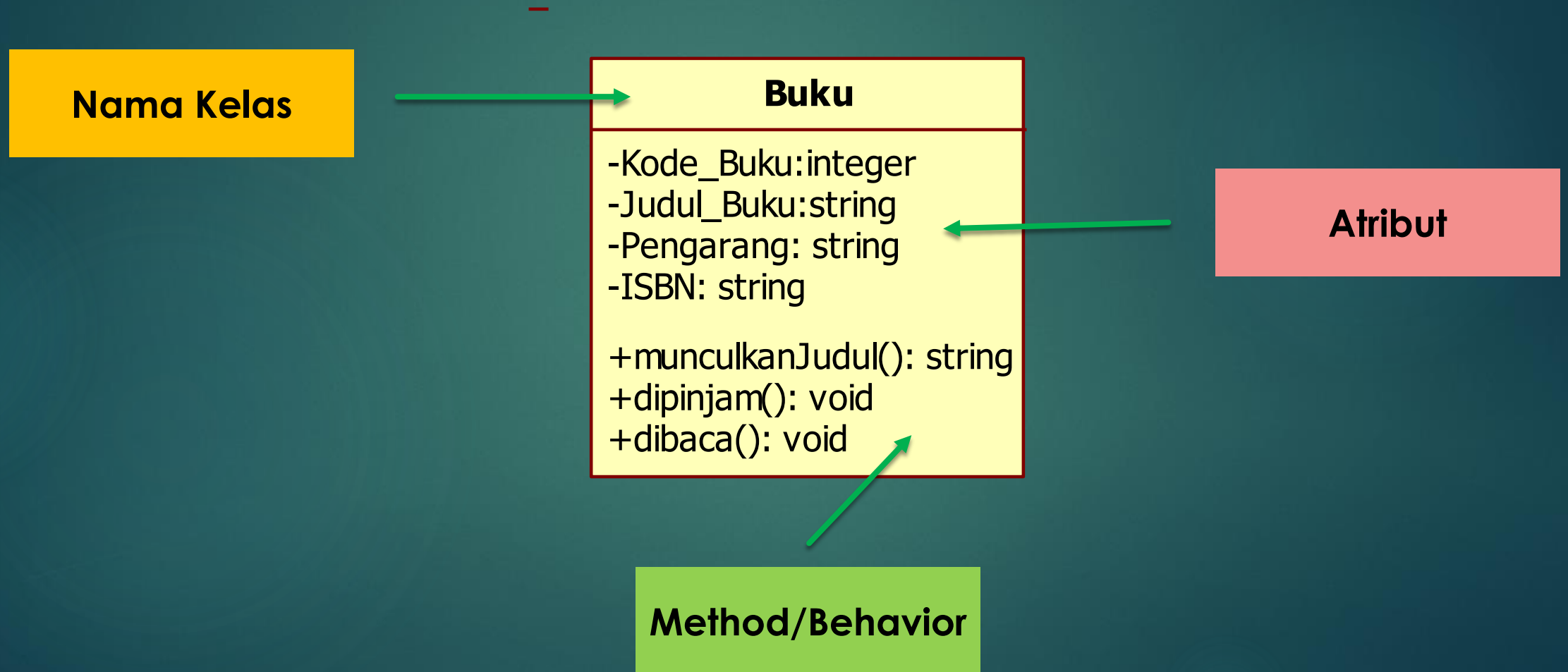
# Definisi Kelas

15

- ① **Sekumpulan objek yang memiliki kemiripan** dalam hal properti, atribut, behavior, dan semantik.
- ② Proses **klasifikasi** dilakukan untuk membentuk kelompok dari beberapa objek yang memiliki kemiripan.

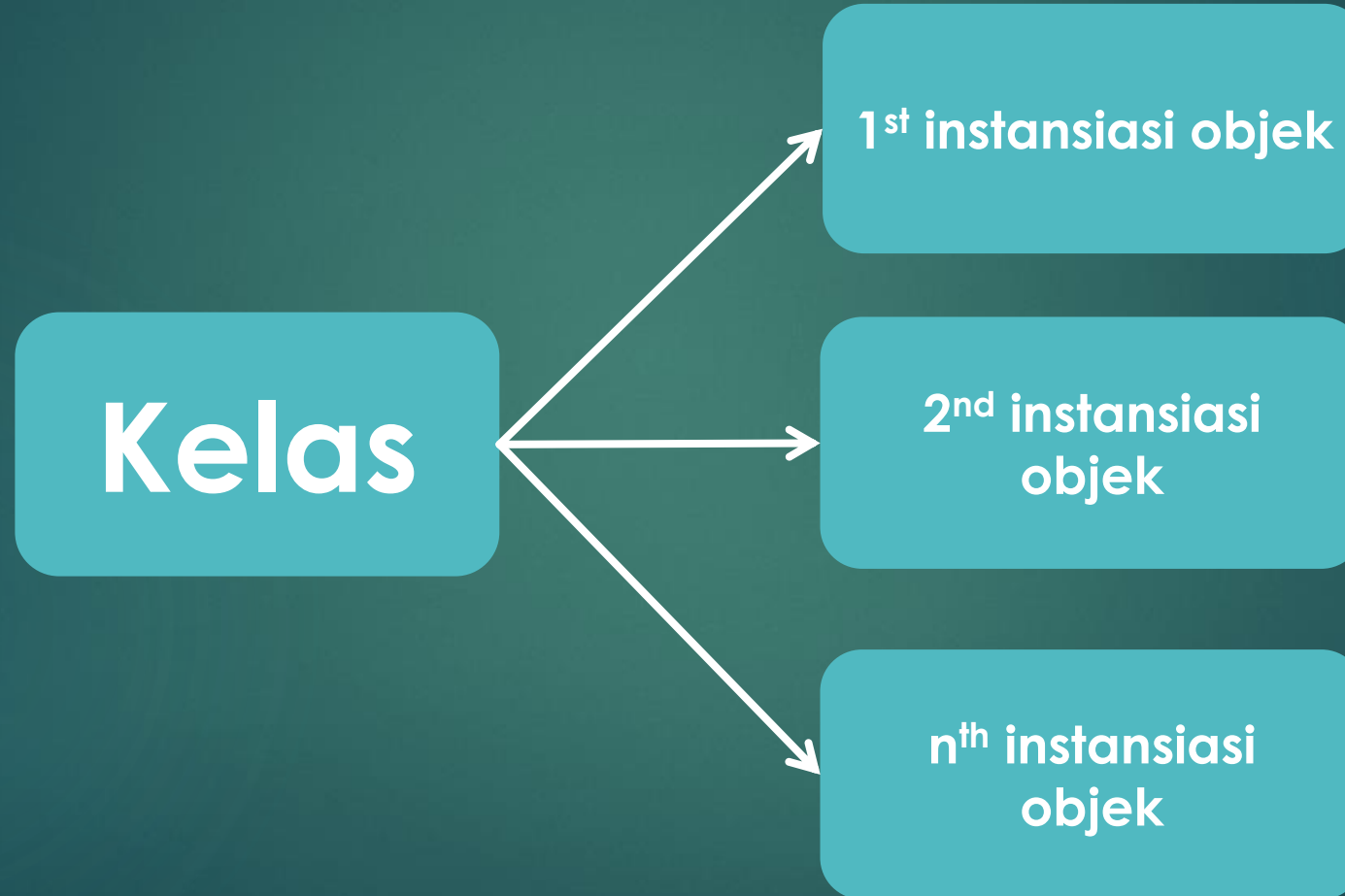
# Ilustrasi Kelas

16



# Objek VS Kelas

17



# OOAD VS OOP

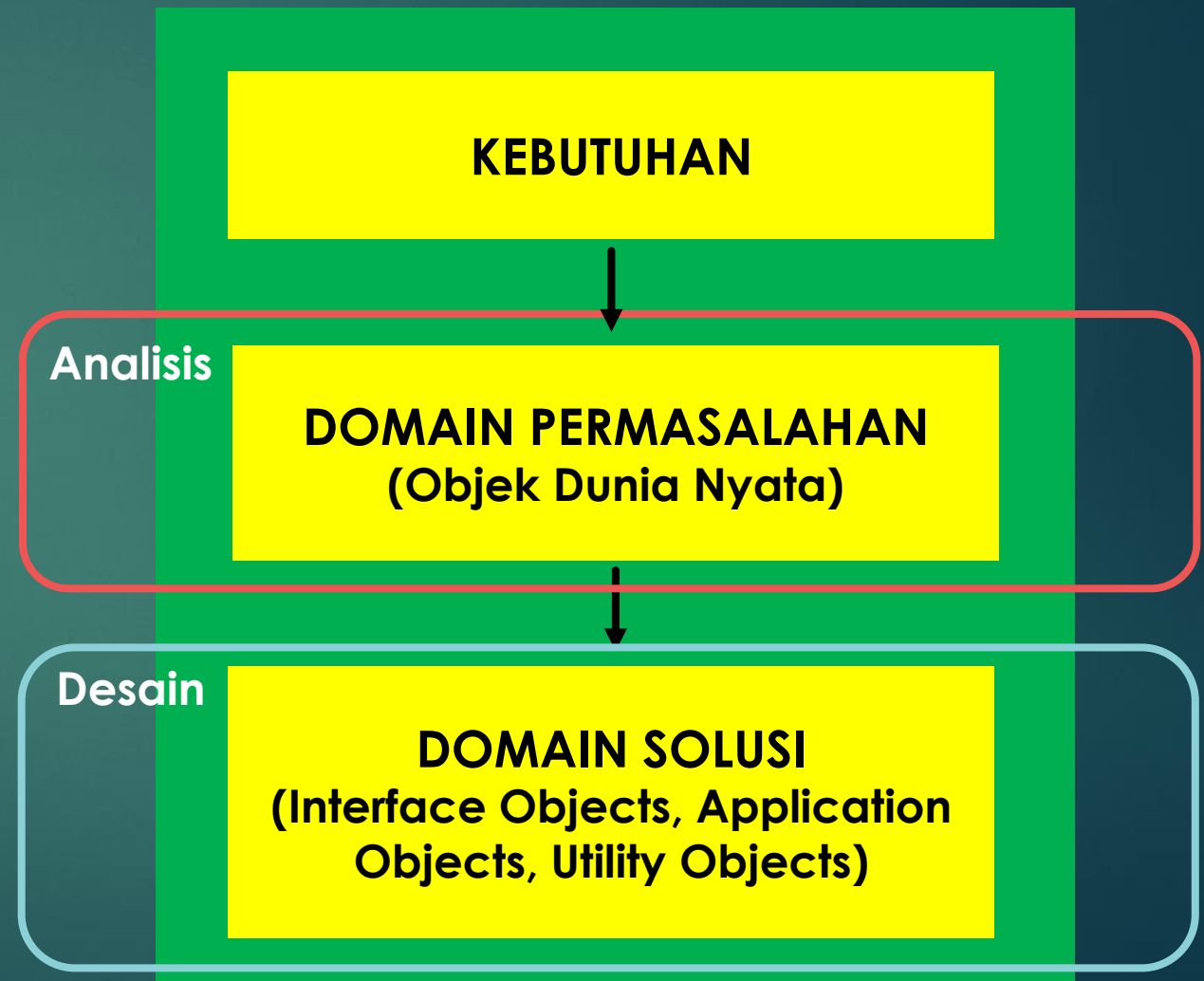
18



# Konsep Dasar OOAD

19

- ① Abstraksi
- ② Encapsulation
- ③ Inheritance
- ④ Polimorphisme



# Definisi Abstraksi

20

- ① Fokus terhadap **esensi**.
- ② **Menghilangkan** sejumlah detail
- ③ Fokus terhadap **“is and does”** dari sebuah objek.



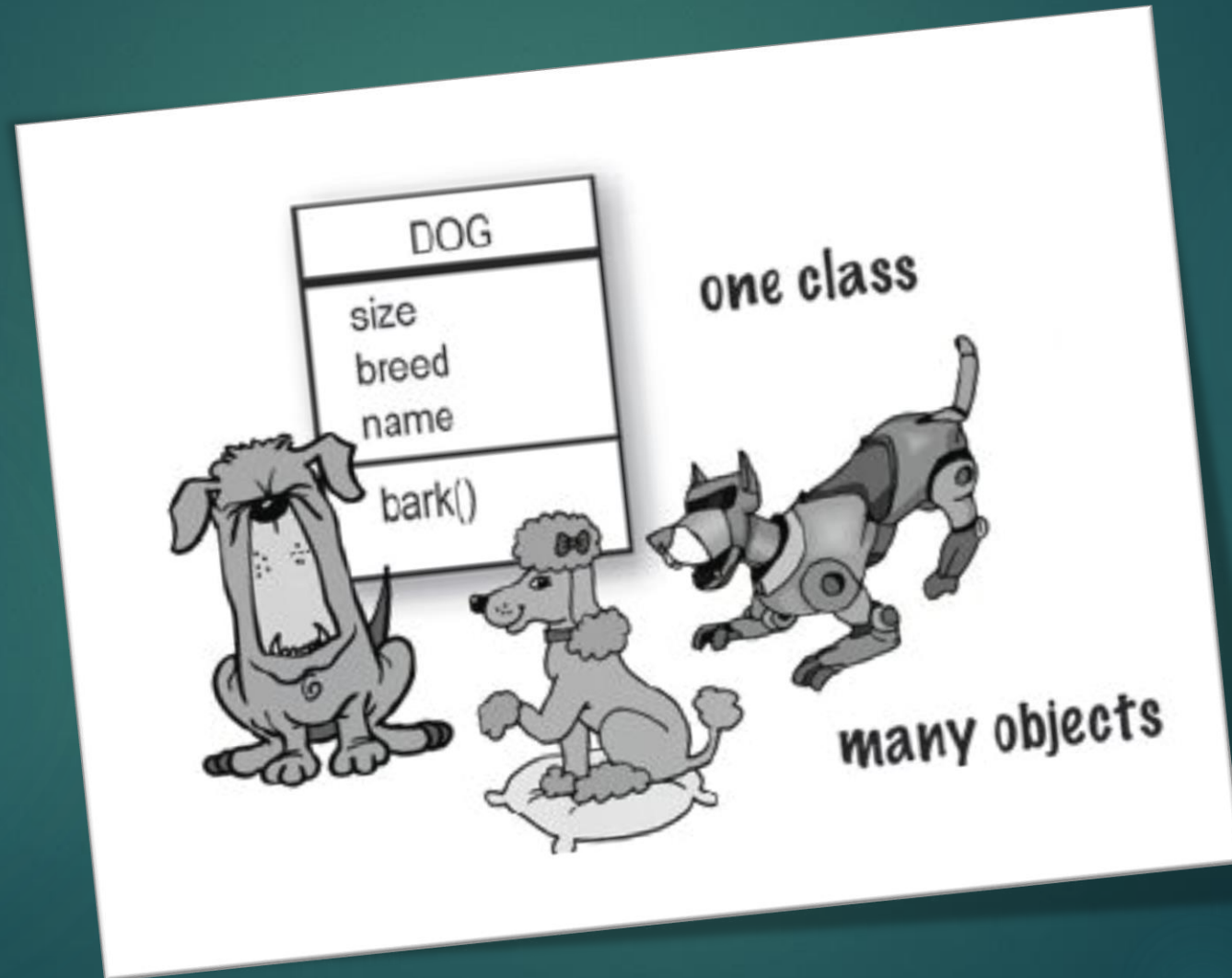
# Definisi Encapsulation

21

- ① Lebih dikenal sebagai **information hiding**.
- ② Melakukan pembungkusan **atribut** dan **method** pada sebuah kelas.

# Ilustrasi Abstraksi dan Encapsulation

22



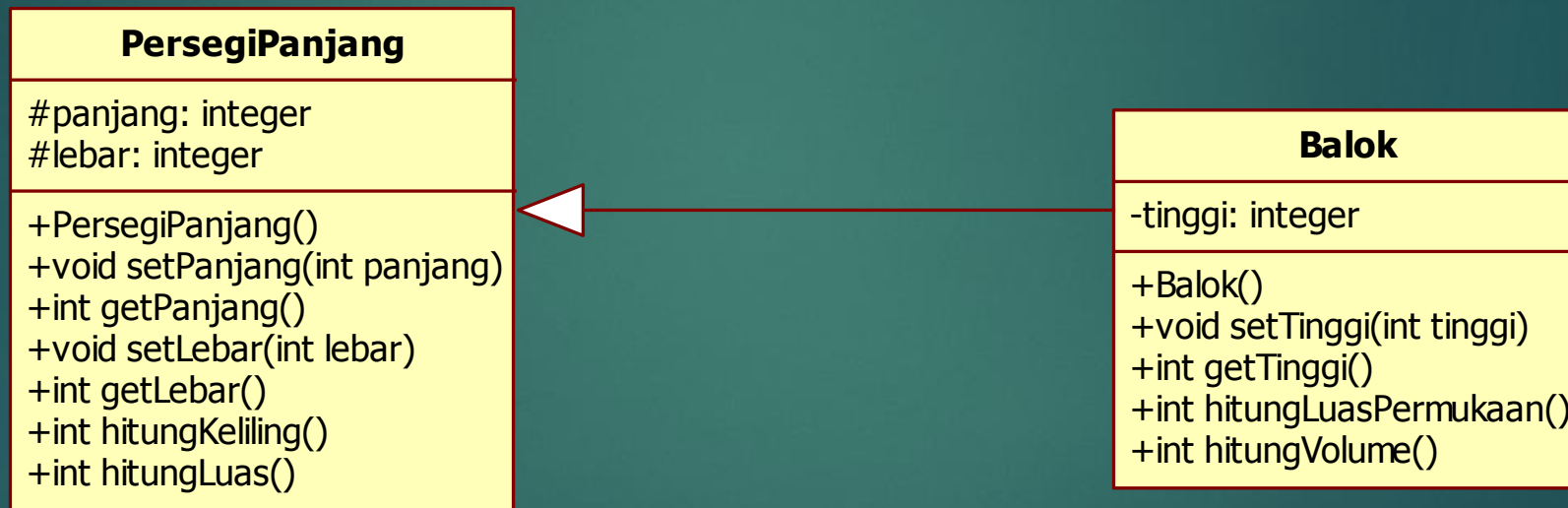
# Definisi Inheritance

23

- ❶ Menyatakan **pewarisan atribut atau method** dari super class ke sub class.
- ❷ Digunakan untuk mendukung konsep **reusable code**.
- ❸ Hak akses dari suatu atribut dan method **memberikan pengaruh** terhadap pewarisan.

# Ilustrasi Inheritance

24



# Definisi Polimorphisme

25

- ① Perbedaan **konten method** yang ada pada dua buah objek atau lebih pada **esensi method yang sama**.
- ② Konsep **overloading** dan **overriding** digunakan sebagai implementasi polimorphisme.
- ③ Sub class diizinkan **mempunyai isi method yang berbeda** dari method super classnya.

# Ilustrasi Polimorphisme

26

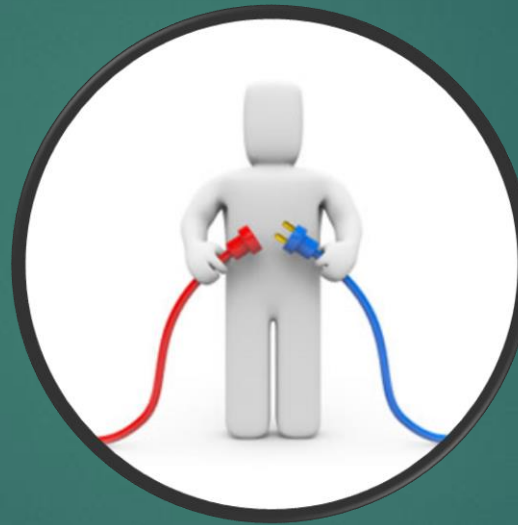




# UML Sebagai Tools OOAD

27

**OOAD**



**UML**

# Sejarah UML

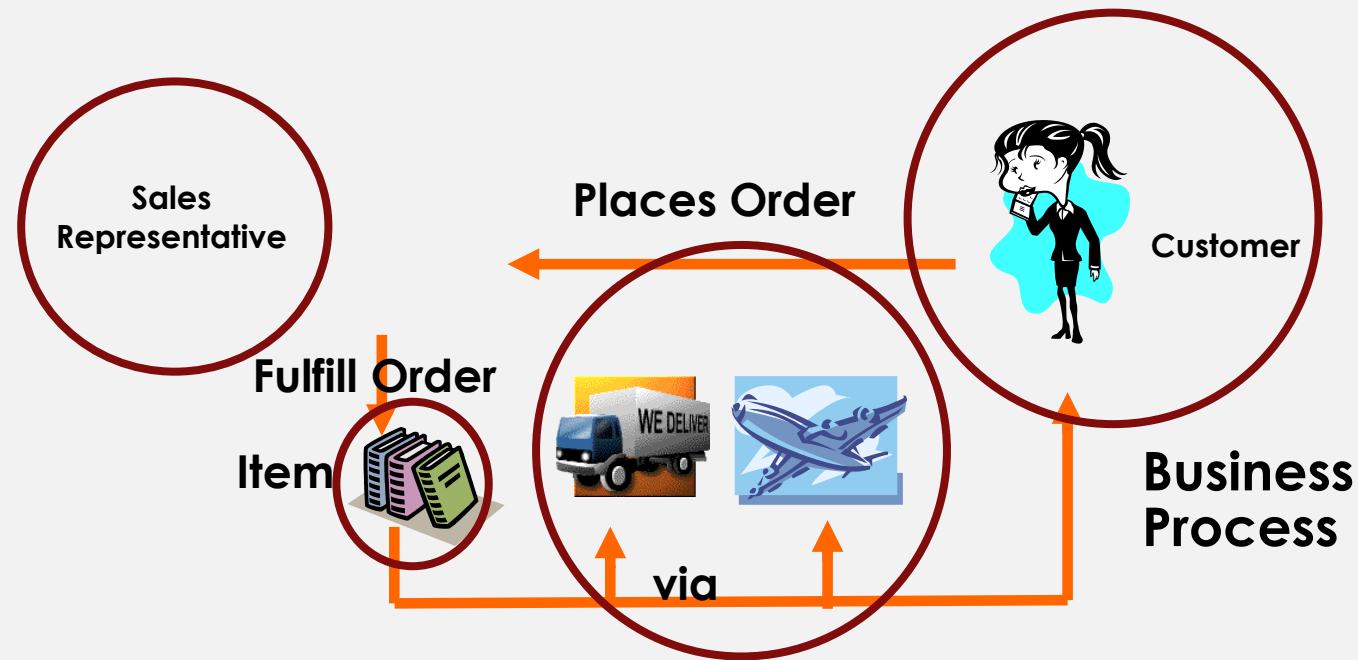
28

- ① OO languages muncul pada pertengahan tahun 70 sampai 80.
- ② Dicituskan oleh “**Three Amigos**” yang terdiri dari Grady Booch, James Rumbaugh, Ivar Jacobson.
- ③ UML digunakan sebagai **alat bantu dalam pemodelan** di OOAD.

# Perbedaan Cara Pandang Analisis

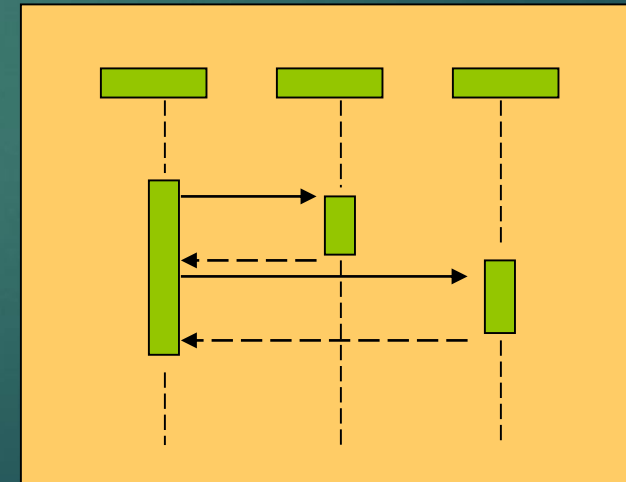
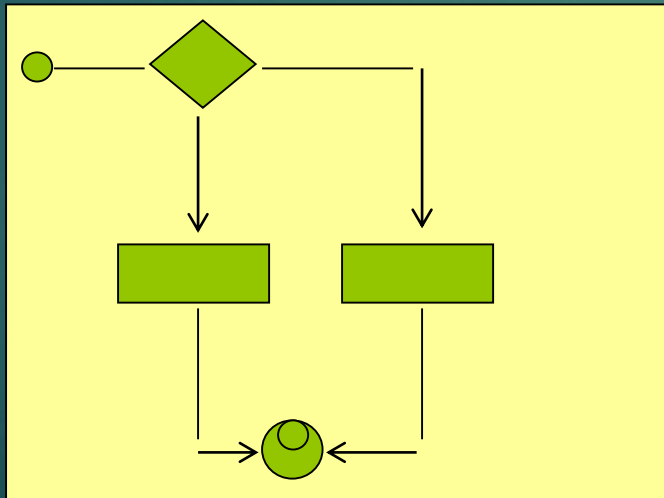
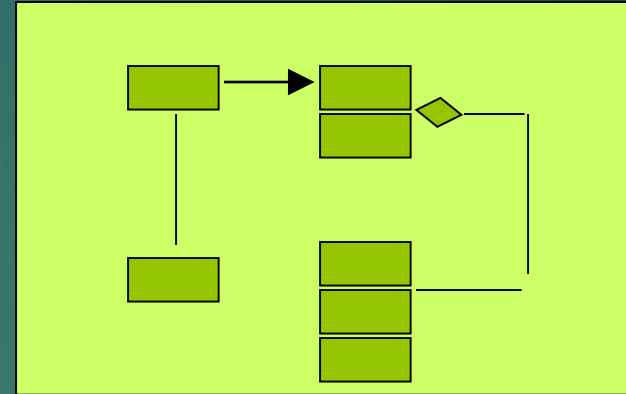
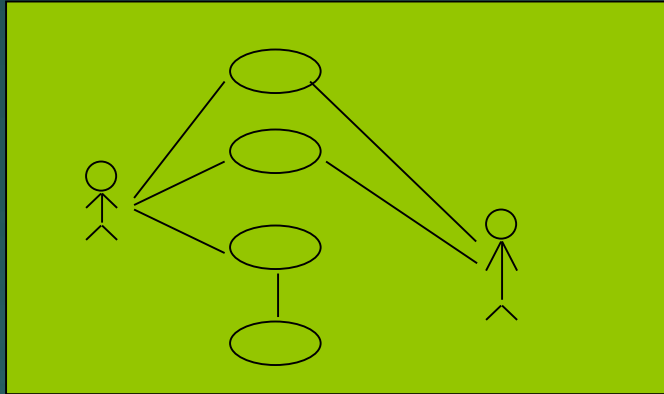
29

## OOAD



# UML Untuk Visual Modelling

30



# UML 1.0 VS UML 2

31

## UML 1.x: 9 diagram types.

### Structural Diagrams

Represent the *static* aspects of a system.

- Class;  
Object
- Component
- Deployment

### Behavioral Diagrams

Represent the *dynamic* aspects.

- Use case
- Sequence;  
Collaboration
- Statechart
- Activity

## UML 2.0: 12 diagram types

### Structural Diagrams

- Class;  
Object
- Component
- Deployment
- Composite Structure
- Package

### Behavioral Diagrams

- Use case
- Statechart
- Activity

### Interaction Diagrams

- Sequence;  
Communication
- Interaction  
Overview
- Timing

# UML 1.0 VS UML 2

32

## UML 1.x: 9 diagram types.

### Structural Diagrams

Represent the *static* aspects of a system.

- Class;  
Object
- Component
- Deployment

### Behavioral Diagrams

Represent the *dynamic* aspects.

- Use case
- Sequence;  
Collaboration
- Statechart
- Activity

## UML 2.0: 12 diagram types

### Structural Diagrams

- Class;  
Object
- Component
- Deployment
- Composite Structure
- Package

### Behavioral Diagrams

- Use case
- Statechart
- Activity

### Interaction Diagrams

- Sequence;  
Communication
- Interaction  
Overview
- Timing



Terima Kasih