



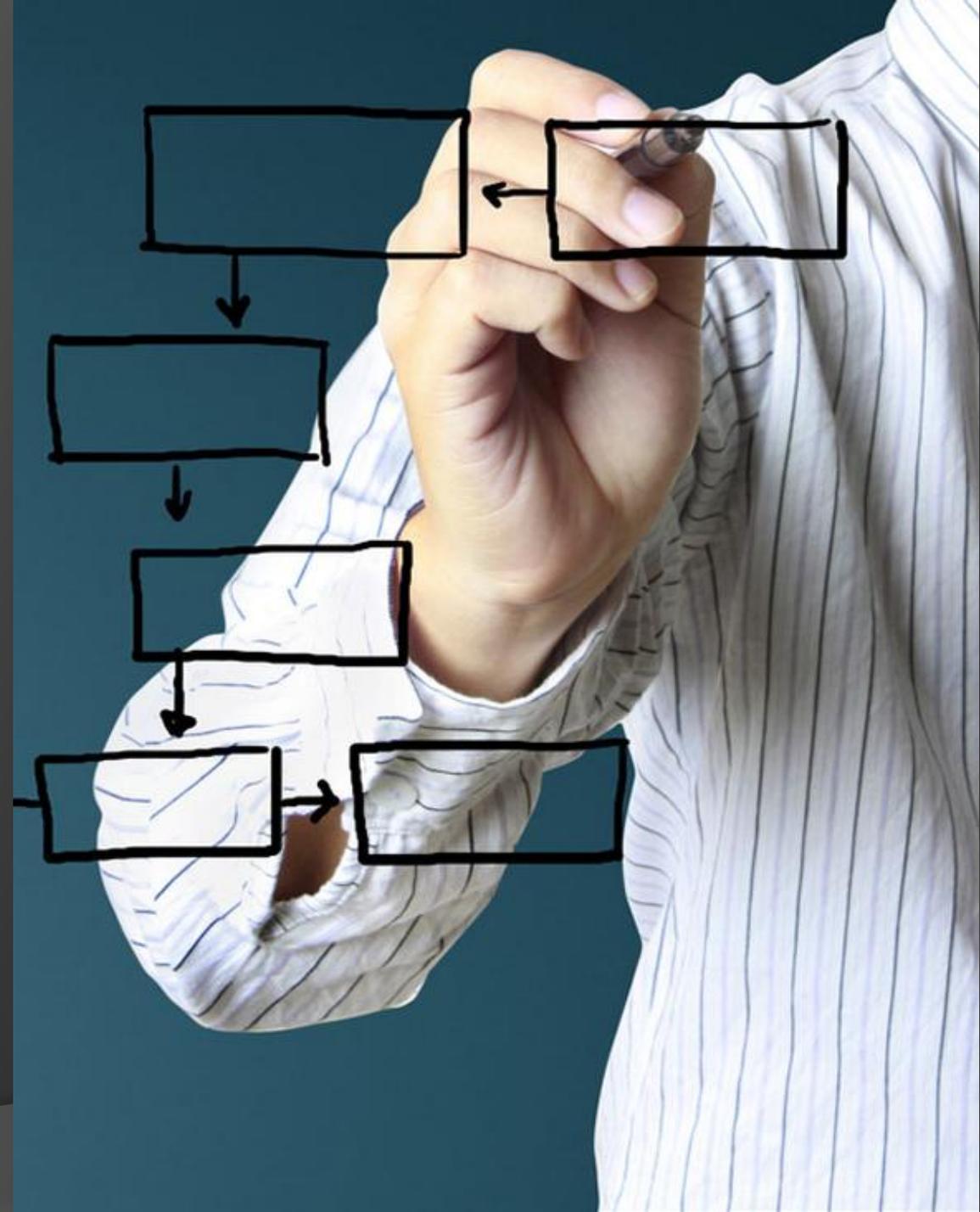
REKAYASA PERANGKAT LUNAK II

Pendahuluan Rekayasa Perangkat Lunak II

Alif Finandhita

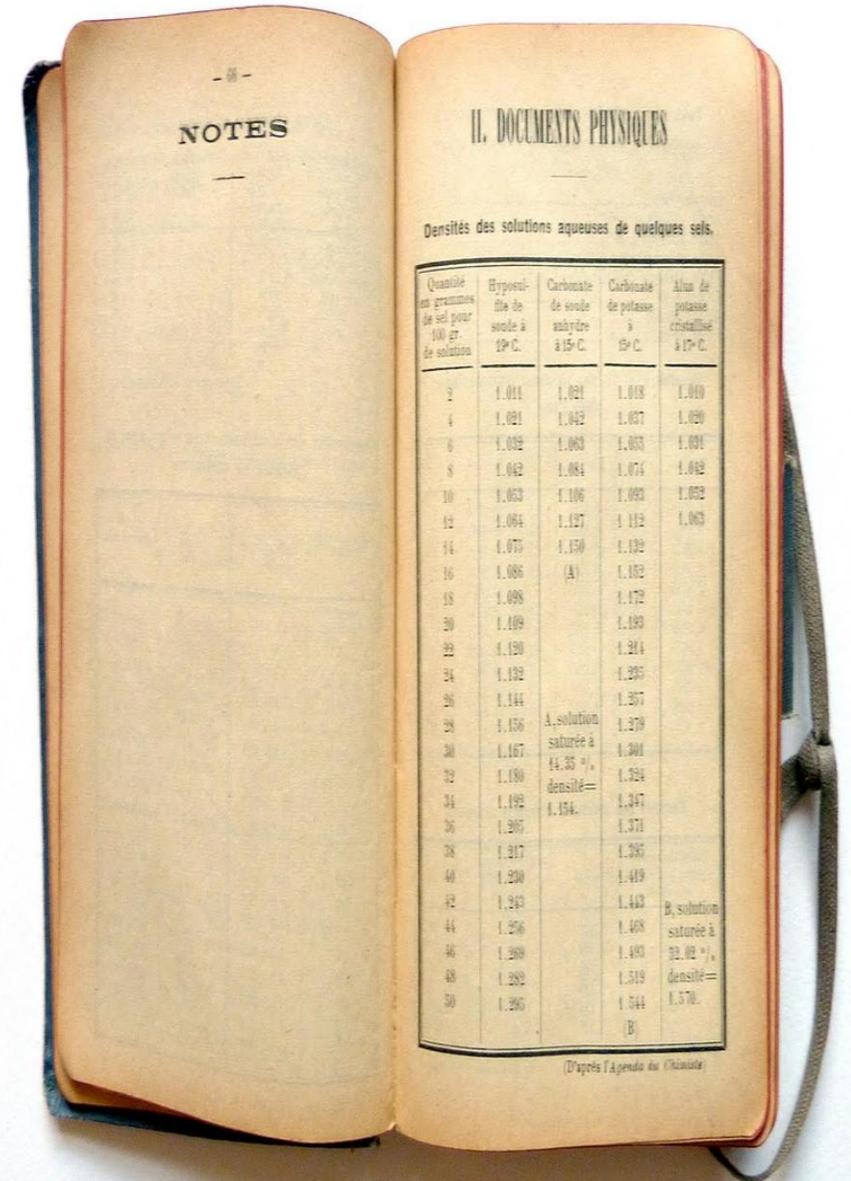
Teknik Informatika UNIKOM

finandhita@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

➡ Sifat : Wajib

➡ Prasyarat : Rekayasa Perangkat Lunak I

➡ Waktu : 3 SKS

Silabus Mata Kuliah

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

Penilaian dan Referensi

Komponen Penilaian

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

Referensi

- ➔ 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

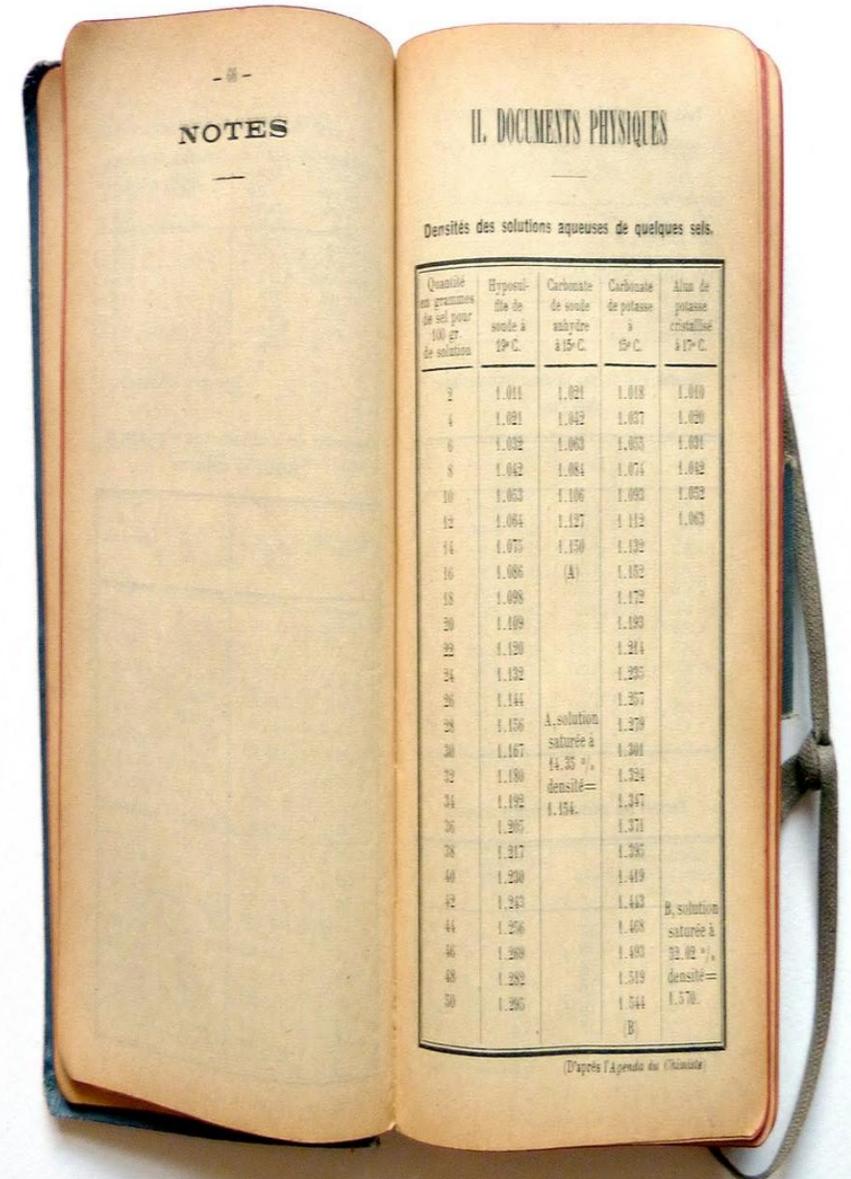
⇒ OOAD

⇒ Objek VS Kelas

⇒ OOAD VS OOP

⇒ Konsep Dasar OOAD

⇒ UML Sebagai tools OOAD



OOAD

- ① Singkatan dari **Object Oriented Analysis and Design**
- ② Salah satu **pendekatan analisis dan desain** yang bisa digunakan selain analisis terstruktur.
- ③ OOAD **bukan dipilih berdasarkan bahasa pemrograman** yang digunakan.
- ④ Pola pikir yang menitik beratkan pada **perekayaan objek beserta relasinya**.

Pengertian OOAD

- ① **Analysis** - Pemahaman, menemukan, dan mendeskripsikan konsep di dalam satu domain masalah.
- ② **Design** - Pemahaman dan pendeskripsian solusi dari perangkat lunak yang merepresentasikan konsep analisis dan akan diimplementasikan pada kode program yang dibuat.
- ③ **OOAD** - Analisis dan Desain menggunakan pendekatan orientasi objek. Sebuah pendekatan pembangunan perangkat lunak yang menekankan pada solusi logika berbasis objek.

Objek



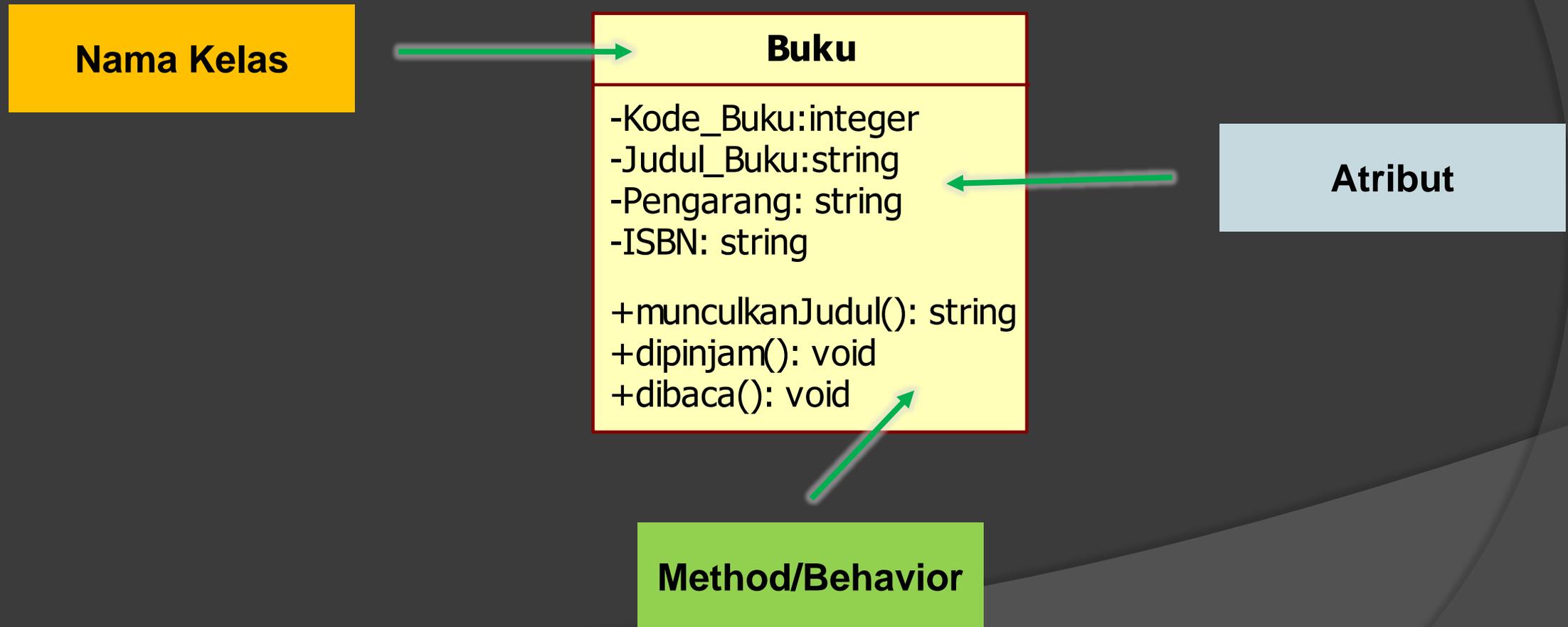
Definisi Objek

- 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.

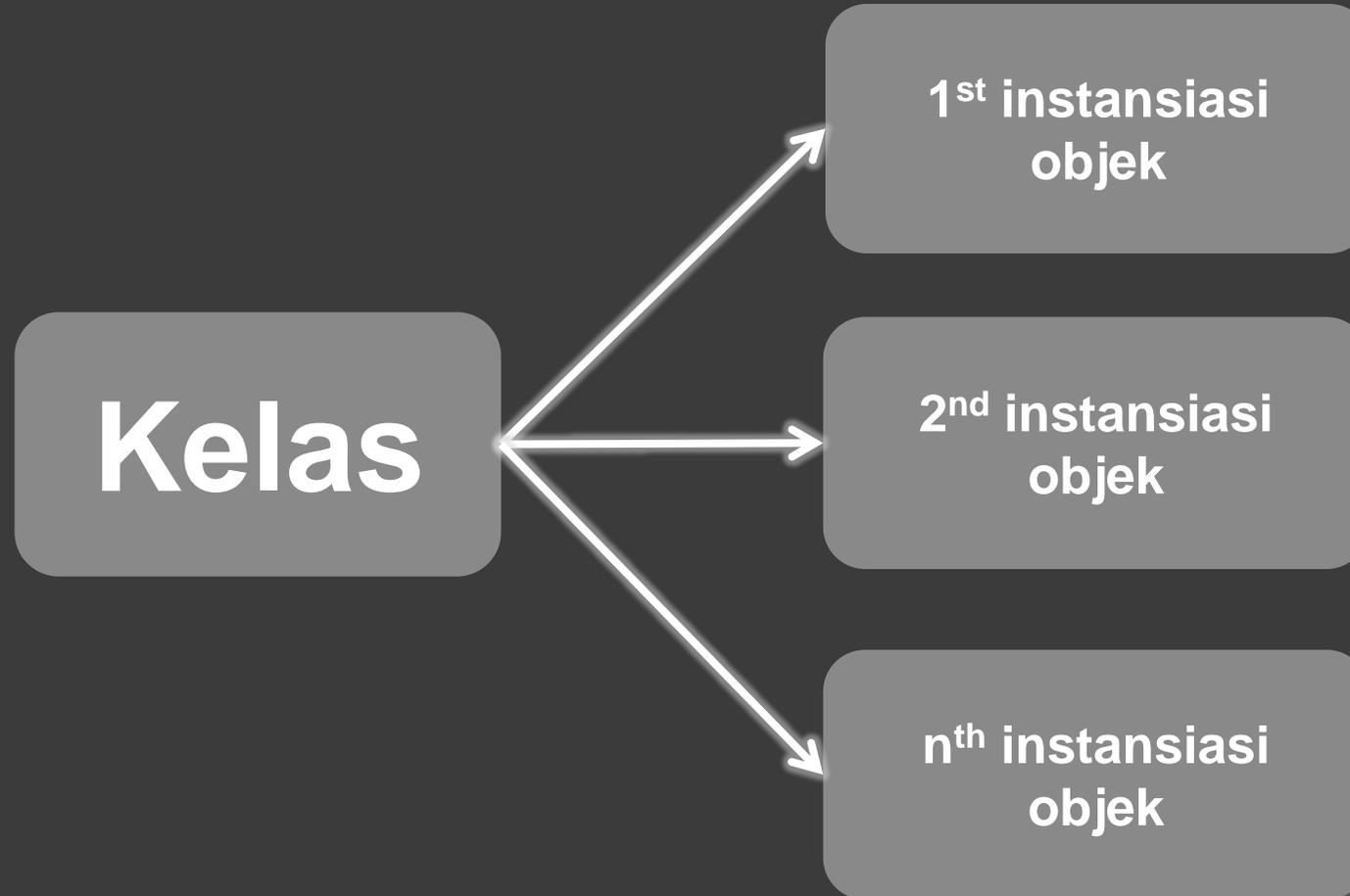
Definisi Kelas

- 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



Objek VS Kelas

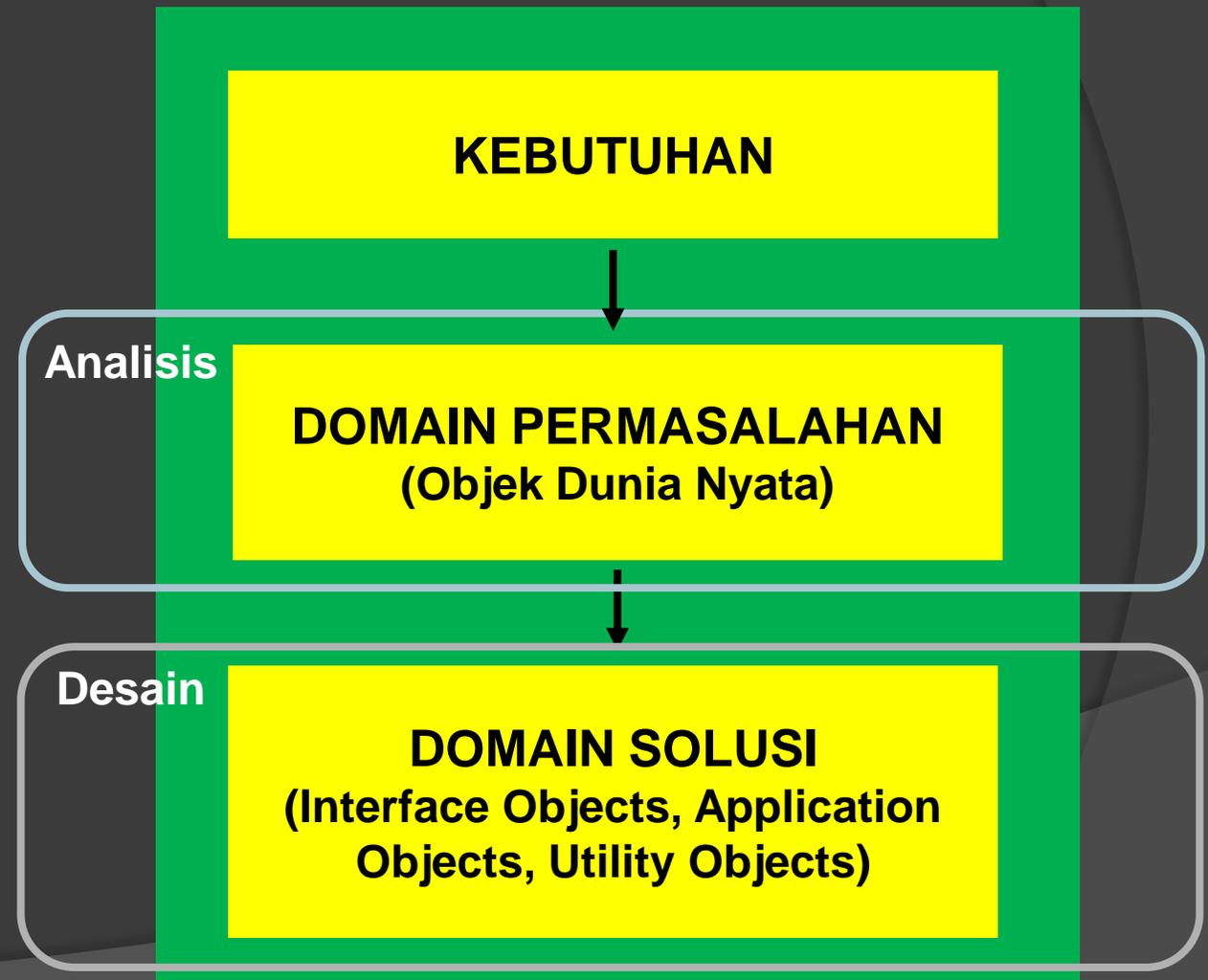


OOAD VS OOP



Konsep Dasar OOAD

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



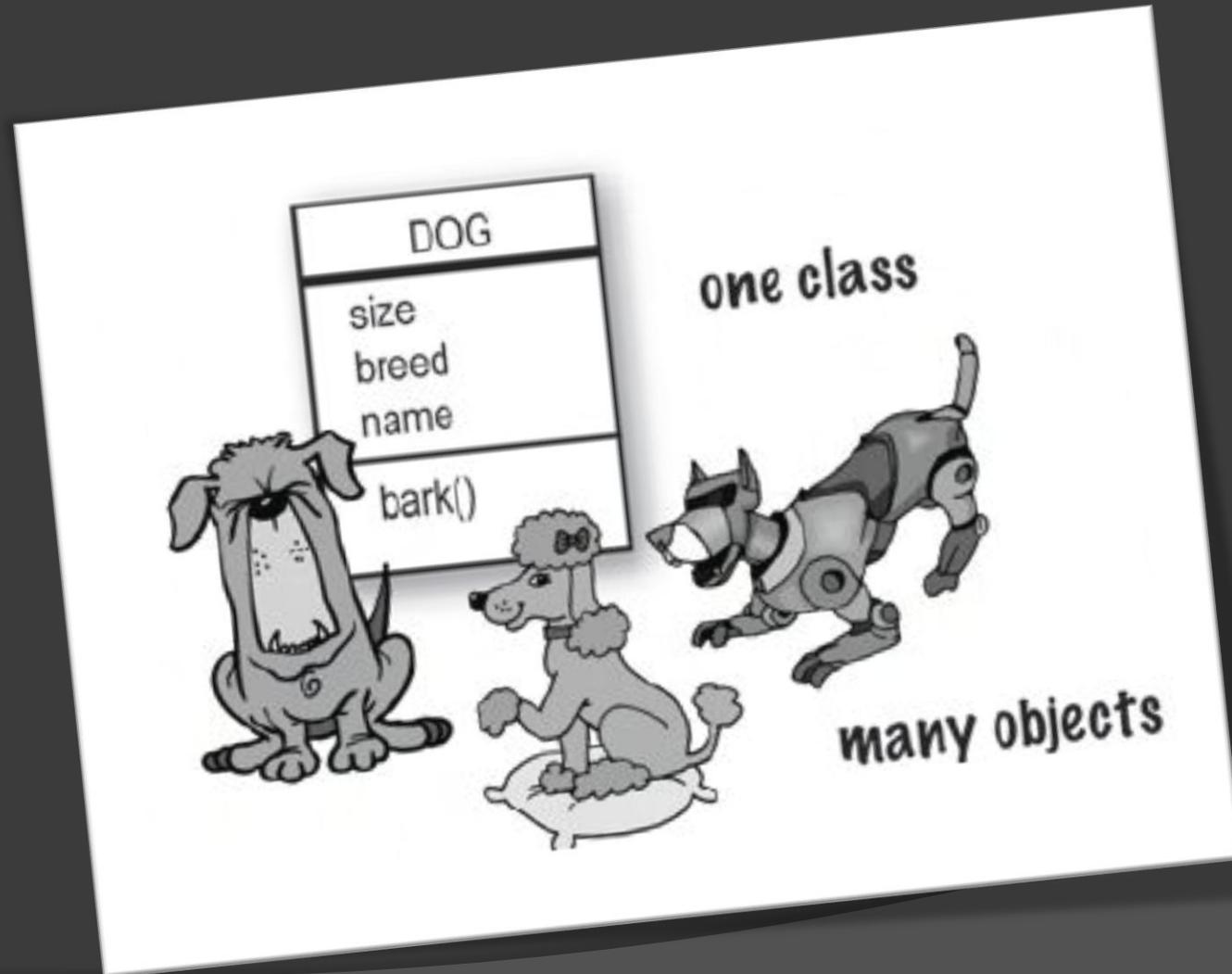
Definisi Abstraksi

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

Definisi Encapsulation

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

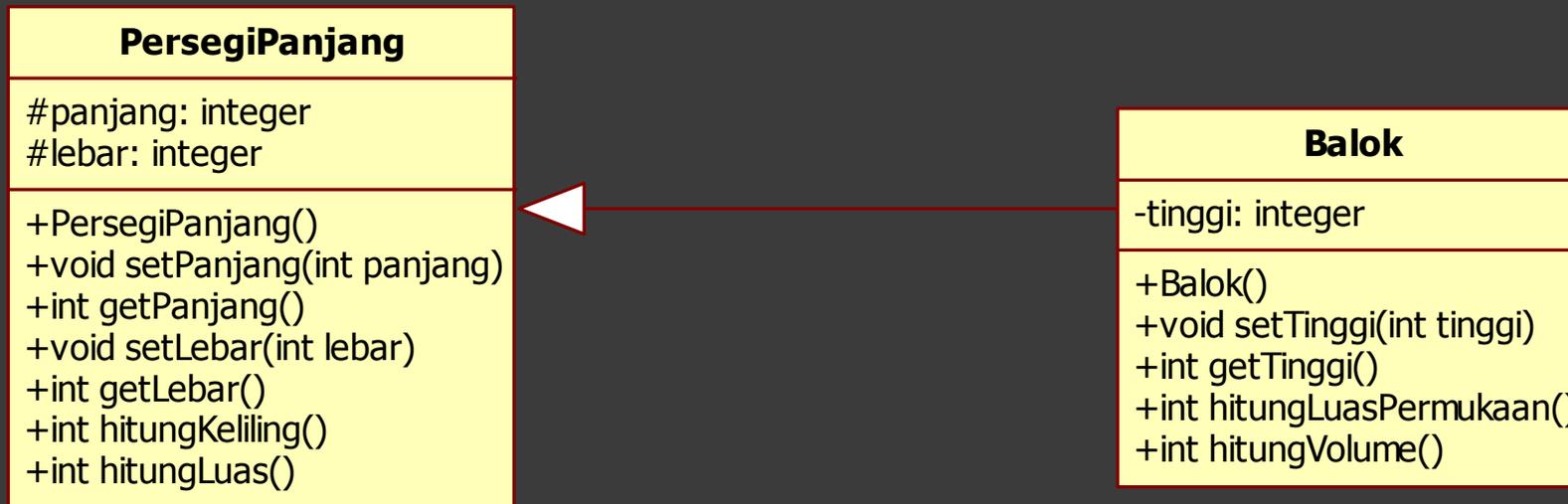
Ilustrasi Abstraksi dan Encapsulation



Definisi Inheritance

- ① Menyatakan **pewarisan atribut atau method** dari super class ke sub class.
- ② Digunakan untuk mendukung konsep **reusable code**.
- ③ Hak akses dari suatu atribut dan method **memberikan pengaruh** terhadap pewarisan.

Ilustrasi Inheritance



Definisi Polimorphisme

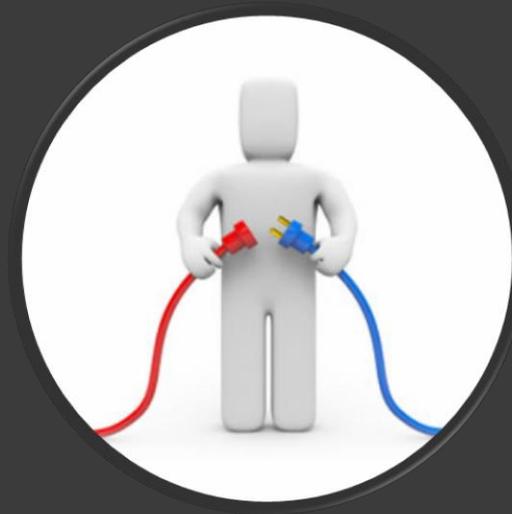
- ① Perbedaan **konten method** yang ada pada dua buah objek atau lebih pada **esensi method yang sama**.
- ② Konsep **overloading** dan **overriding** digunakan sebagai implementasi polimorphisme.
- ③ Sub class diizinkan **mempunyai isi method yang berbeda** dari method super classnya.

Ilustrasi Polimorphisme



UML Sebagai Tools OOAD

OOAD



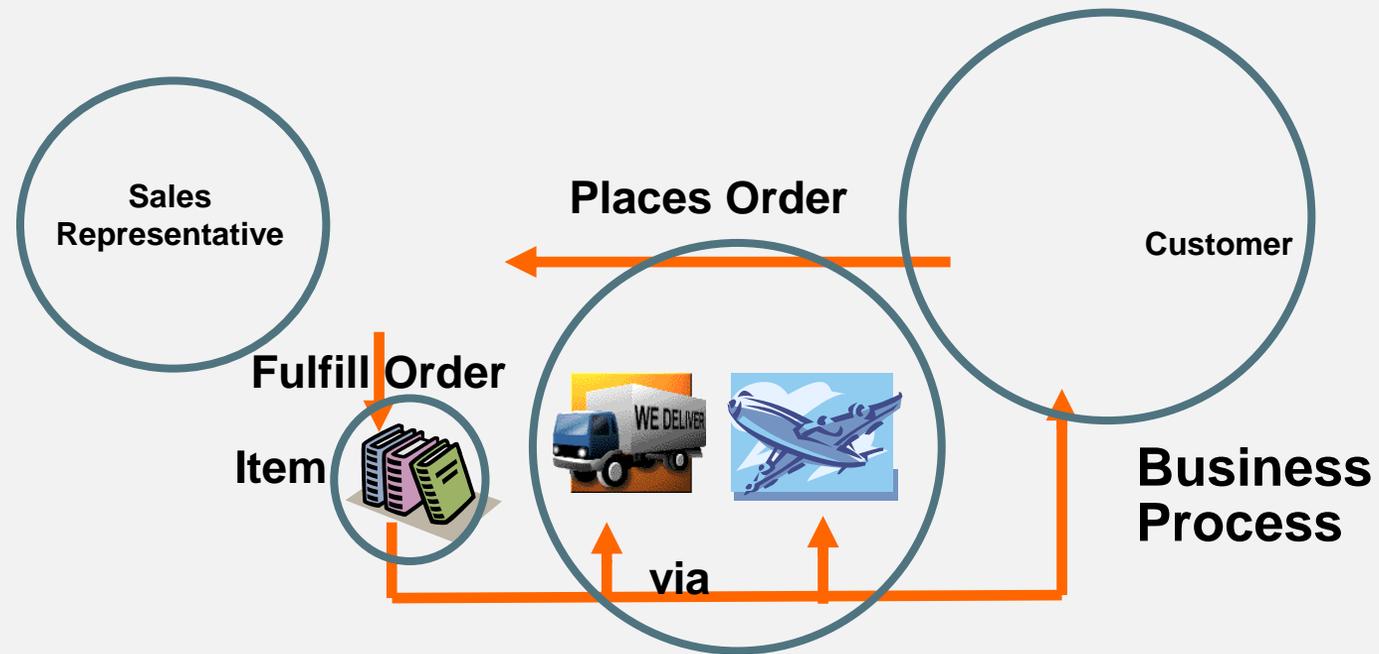
UML

Sejarah UML

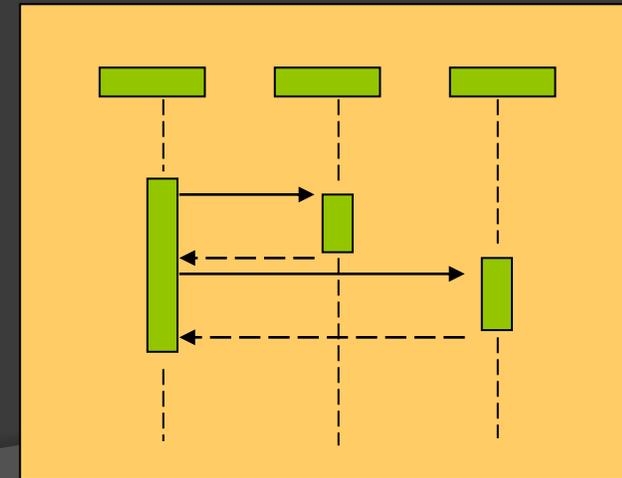
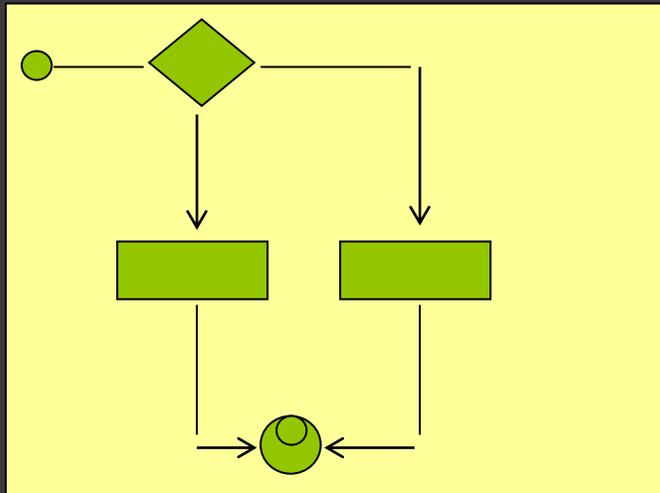
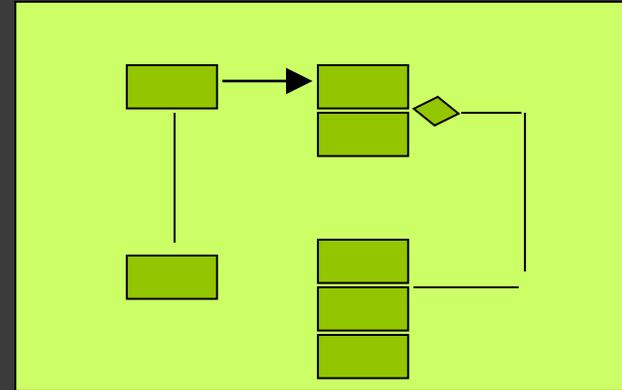
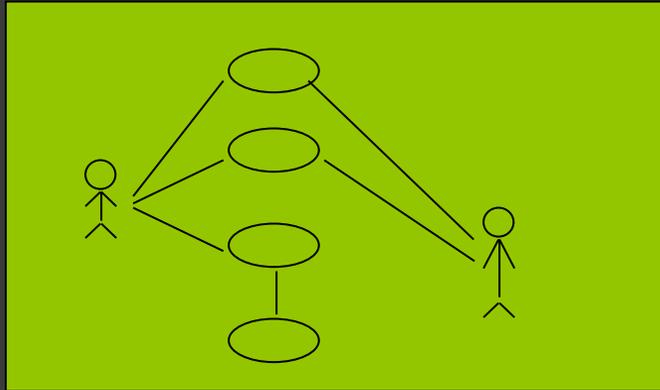
- ① OO languages muncul pada pertengahan tahun 70 sampai 80.
- ② Dicituskan oleh “**Three Amigos**” yang terdiri dari Grady Booch, James Rumbaugh, Ivar Jacobson.
- ③ UML digunakan sebagai **alat bantu dalam pemodelan** di OOAD.

Perbedaan Cara Pandang Analisis

OOAD



UML Untuk Visual Modelling



UML 1.0 VS UML 2

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 Interaction Diagrams

- Use case
- Statechart
- Activity

- Sequence;
Communication
- Interaction
Overview
- Timing

Terima Kasih