

# Object Oriented Methodology (Concept)

Inggriani Liem

# Dua Pendekatan Utama

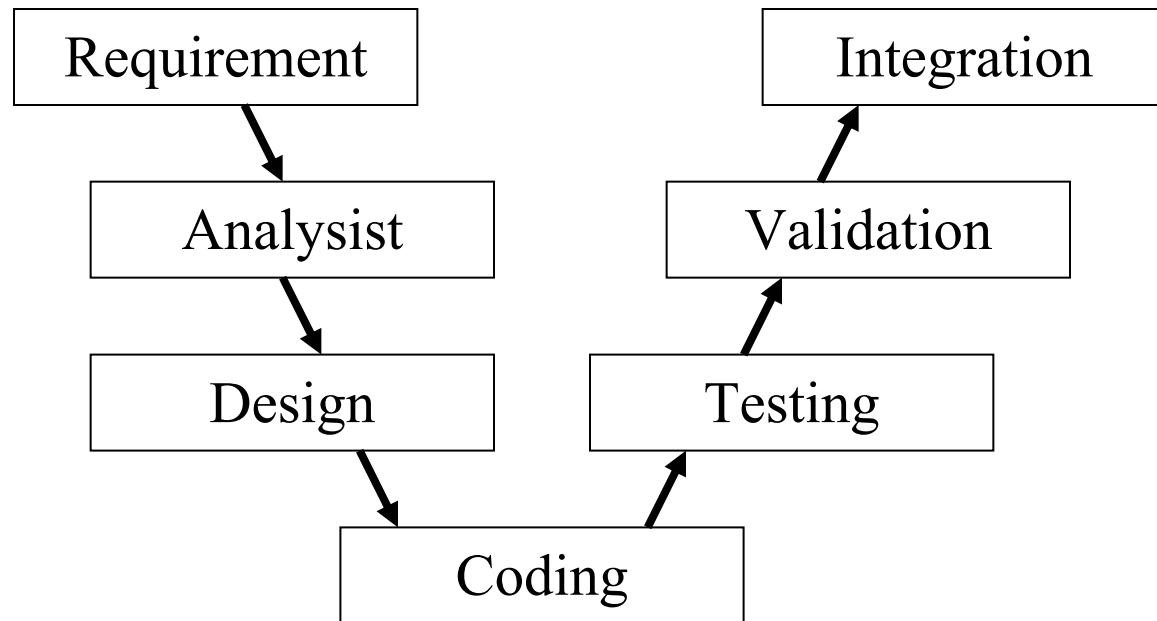
- **Functional:** pendekatan dimulai dari fungsi, diidentifikasi fungsi software yang diperlukan untuk mendukung proses bisnis
- **Object :** pendekatan dimulai dari “objek”, diidentifikasi objek/benda apa yang menjadi tanggung jawab dan harus dikelola oleh software.

# Metodologi Fungsional

- SA/SD - Structured Analysis, Structured Design :
  - Eduard Jordon
  - Tom de Marco
  - Gane Sarson
  - Jacobson
  - SSADM (Goodland)
- SART:tambahan aspek Real Time pada SA

# OO... dan siklus hidup PL

- OOA
- OOD
- OOP
- OOT



# OO & Life Cycle

- OOA: analisis persoalan dengan salah satu methodologi tertentu. Hasil utamanya adalah sekumpulan diagram sesuai dengan metodologi yang dipakai. “Sentra” dari produk adalah SW Use case, Diagram Class (dengan relasi, asosisasi antar kelas), dan diagram yang menyatakan dinamika (behaviour, kelakuan, ..) dan interaksi dengan user
- OOD: buat detail design, skema solusi secara keseluruhan
- OOP: implementasi setiap kelas (jika belum) dalam salah satu bahasa pemrograman yang dipilih dengan/tanpa memanfaatkan library yang tersedia.
- OOT:
  - unitary test: test setiap kelas yang independent
  - kemudian test semua kelas yang mempunyai hubungan.
  - Test seluruh sistem (lebih sulit jika ada banyak objek aktif)
- Sarana reusability: class library, design pattern

# Metodologi OO

- Ada banyak sekali, pelajari a.l.
    - OOA, OOD -COAD & Jordon
    - OMT- Rumbaugh
    - Booch
    - Jacobson
    - UML *[catatan: di beberapa literatur disebutkan bahwa UML bukan metodologi, melainkan “Language”]*
- Notasi
  - Diagram
  - Bahasa Spesifikasi
  - Case Tools

**Bacalah:** [http://www.cetus-links.org/oo\\_ooa\\_ood\\_methods.html](http://www.cetus-links.org/oo_ooa_ood_methods.html)

# Beberapa Metodologi

- OMT-Rumbaugh
  - Object modeling - Obejct diagram
  - Dynamic Modeling - State diagram
  - Functional Modeling - DFD
- Booch
  - Class Diagram
  - State transition Diagram
  - Object Diagram
  - Timing Diagram
  - Module Diagram
  - Process Diagram

# UML

- Use Case driven
- Architecture-Centric Process
- Iterative and Incremental process
- Models: use-case, design model, implementation model



# UML-diagram

- **Use-case**
- **Class, object diagram**
- Sequence diagram
- Collaboration diagram
- State chart diagram
- Activity diagram
- Component diagram
- Deployment diagram

# Catatan Mengenai UML

- UML adalah sebuah bahasa yang dipakai pada seluruh life cycle
- User dapat membuat “stereotype” untuk kepentingan tertentu, namun harap memperhatikan stereotype yang sudah dipublish untuk domain tertentu, misalnya Web application, real-time system,...

# UML : Use Case

- Perhatikan penamaan Use case:
  - Taatilah naming convention
  - Kata kerja (verbs), diikuti kata benda
  - Setiap kata kerja yang sama harus mengandung “task” yang sama
  - Contoh : create transaction, send email, ...
- Diagram mana yang dibuat ?
  - Biasanya Use case/scenario, namun tidak selalu demikian
- Harap Dibedakan :
  - System Use Case/scenario
  - Software Use Case/scenario

# UML : Class Diagram

- Class Diagram :
  - dari class diagram, dapat digenerate source code dalam bahasa tertentu, hasilnya sebuah “skeleton” yang harus dilengkapi kodenya
  - Memasuki tahapan “analisis”, mulai digambarkan sketsanya, pada saat design di “refine”.
  - Seringkali, jika functional specification sudah jelas, maka technical spec dimulai dari class diagram
  - Perhatikan penamaan kelas : Noun (biasanya untuk ADT, atau “Verbal Noun” (pemroses Objek), atau Contoller. Contoh : Point, PointMover, PointManager

# UML

## Sequence & Collaboration Diagram

- Sequence & Collaboration Diagram dibangkitkan dari class diagram
- Sequence diagram :
  - message passing & exchange antar object
  - Dari sequence diagram dapat dibangkitkan kode yang sesuai (invokasi, message passing)
- Collaboration diagram :
  - Kolaborasi antar kelas
  - Abstraksi lebih tinggi

# Pustaka

- [Meyer97] Bertrand Meyer, “Object Oriented Software Construction”, 2<sup>nd</sup> edition, Prentice Hall, 1997.
- [Booch95] Grady Booch, “Object oriented Analysis and Design with Application”, The Benjamin/Cumming Publishing Company, 1995
- [Jacobson192] Jacobson, Ivar: “Object Oriented Software Engineering”, Addison Wesley, 1992
- [UML99] Jacobson, Booch , Rumbaugh : “The Unified Software Development Process”, Addison Wesley, 1999

# Pustaka

- [Rumbaugh91] Rumbaugh, Blaha, Premerlani, Eddy, Lorensen : “Object Oriented Modeling and Design”, Prentice Hall, 1991
- [Martin95] Martin, Odell : “Object Oriented Methods”, A Foundation, Prentice Hall, 1995
- [Coad91] Coad, Yourdon : “Object Oriented Analysis”, Yourdon Press, 1991
- [Coad91] Coadd, Yourdon : “Object Oriented Design”, Yourdon Press, 1991