Tim Pengajar IF2250

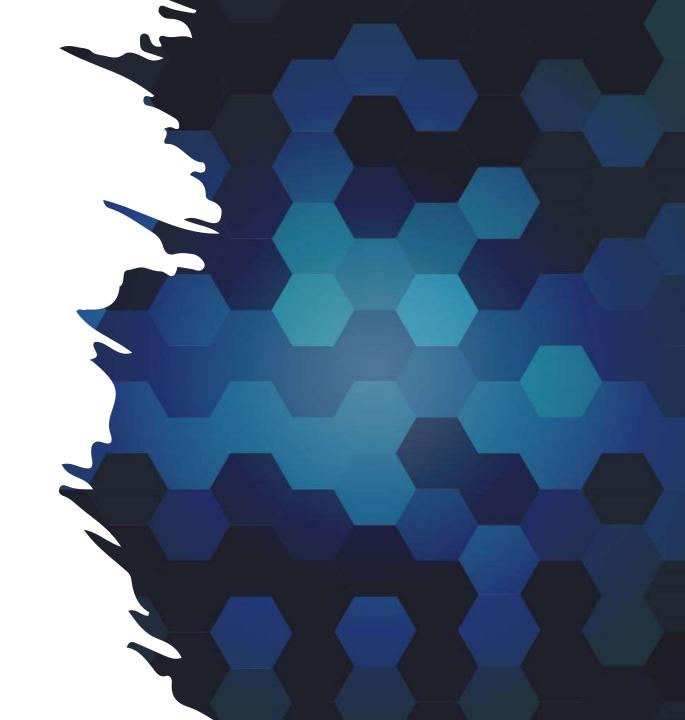
IF2250 – Rekayasa Perangkat Lunak

OO Methods (Bag 1)

SEMESTER II TAHUN AJARAN 2022/2023







Object Oriented methodologies

- A. The Rumbaugh OMT
- B. The Booch methodology
- C. Jacobson's methodologies





A. Rumbaugh's Object Modeling Technique (OMT)

- A method for <u>analysis</u>, <u>design</u>, and <u>implementation</u> by an objectoriented technique.
- <u>fast</u> and <u>intuitive</u> approach for <u>identifying</u> and <u>modeling</u> all objects making up a system.
- Class <u>attributes</u>, <u>methods</u>, <u>inheritance</u>, and <u>association</u> can be expressed <u>easily</u>.
- <u>Dynamic behavior</u> of objects can be described using the <u>OMT</u> <u>dynamic model</u>.
- Detailed specification of <u>state transitions</u> and their <u>descriptions</u> within a system

Four phases of OMT (can be performed iteratively)

- 1. Analysis: objects, dynamic, and functional models
- 2. System Design: basic architecture of the system.
- **3. Object Design**: static, dynamic, and functional models of objects.
- 4. Implementation: reusable, extendible, and robust code.





Three different parts of OMT modeling

- 1. An **object model** object model & data dictionary
- 2. A **dynamic model** state diagrams & event flow diagrams
- 3. A **functional model** data flow & constraints





I. Object Model

- structure of objects in a system.
- Identity, relationships to other objects, attributes, and operations.
- Object diagram
 - Classes interconnected by <u>association lines</u>
 - Classes <u>a set of</u> individual objects
 - Association lines relationship among classes (i.e., objects of one class to objects of another class)





2. Dynamic Model

- States, transitions, events, and actions
- OMT state transition diagram network of states and events





3. Functional Model

- DFD (Data Flow Diagram)
- Shows <u>flow of data</u> between <u>different processes</u> in a business.
- Simple and intuitive method for describing <u>business</u> <u>processes</u> <u>without</u> focusing on the details of <u>computer</u> <u>systems</u>.





B. The Booch Methodology

- Widely used OO method
- Uses the object paradigm
- Covers the <u>design</u> and <u>analysis</u> phase of an OO system
- Criticized for his <u>large set</u> of symbols





Diagrams of Booch method

- Class diagrams describe roles and responsibilities of objects
- Object diagrams describe the desired <u>behavior</u> of the system in terms of scenarios
- State transition diagrams state of a class based on a stimulus
- Module diagrams to map out where each class & object should be declared
- Process diagrams to determine to <u>which processor</u> to allocate a process
- Interaction diagrams describes <u>behavior</u> of the system in terms of <u>scenarios</u>





Booch method prescribes:

- Macro Development Process
 - Controlling framework for the micro process.
 - Primary concern-technical management of the system.
 - Each macro process has its own micro development process
- Micro Development Process

Steps:

- Identify classes & objects
- Identify class & objects semantics
- Identify class & object relationship
- Identify class & objects interface and implementation





Steps for macro development process

- 1. Conceptualization
- Analysis & Development of the model
- 3. Design or create the system architecture
- 4. Evolution or implementation
- 5. Maintenance





C. Jacobson Methodologies

- Use Cases.
- Object Oriented Software Engineering (OOSE).
- Objectory is built models
 - Use case model
 - Domain object model
 - Analysis object model
 - Implementation model
 - Test model

- Object Oriented Business Engineering (OOBE)
- OOBE is object modeling at the enterprise level.
 - Analysis phase
 - Design and Implementation phase
 - Testing phase
 - E.g. Unit testing, integration, and system testing.





References

- Object-oriented Modeling and Design by James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, and William Lorensen, Prentice Hall, 1991
- Object-Oriented Analysis and Design with Applications 3rd Edition by <u>Grady Booch</u> (Author), <u>Robert A. Maksimchuk</u> (Author), <u>Michael W. Engle</u> (Author), <u>Bobbi J. Young</u> (Author), <u>Jim Conallen</u> (Author), <u>Kelli A. Houston</u> (Author), Addison-Wesley, 2007
- Object-Oriented Software Engineering: A Use Case Driven Approach 1st edition by <u>Ivar Jacobson</u>, Magnus Christerson, Patrik Jonsson & Gunnar Overgaard. Addison-Wesley, 1992



