

Bab 14: Perancangan dan Pemodelan Berorientasi Obyek

Analisis dan Perancangan Sistem informasi

Monica A. Kappiantari - 2009

Source: Whitten, J.L., L.D. Lonnie and K.C. Dittman, Systems Analysis and Design Methods, 6th ed., McGraw-Hill, Boston, 2004.

Bab 14

Referensi

- Whitten, bab 18

Topik

1. Pendahuluan: Perancangan Berorientasi Obyek
2. Proses Perancangan Obyek

Object-oriented Design (OOD)

- **Object-oriented design (OOD)** – pendekatan yang digunakan untuk menetapkan solusi software dalam kolaborasi dengan obyek, atribut, dan metodenya. Merupakan kelanjutan dari object-oriented analysis
- Proses OOD:
 1. *Refining the use case model to reflect the implementation environment.*
 2. *Modeling object interactions and behavior that support the use case scenario.*
 3. *Updating the object model to reflect the implementation environment.*

1. Refining The Use Case Model

Step 1: Transforming the “Analysis” Use Cases to “Design” Use Cases

Step 2: Updating the Use Case Model Diagram and Other Documentation to Reflect any New Use Cases

2. Modeling Object Interactions and Behaviors

Step 1: Identify and Classify Use-Case Design Objects

Step 2: Identify Object Attributes

Step 3: Model High-Level Object Interactions

Step 4: Identify Object States, Behaviors, and Responsibilities

Step 5: Model Detailed Object Interactions

3. Updating Object Model to Reflect Implementation Environment

Design class diagram – a diagram that depicts classes that correspond to software components that are used to build the software application. Includes:

- Classes
- Associations and gen/spec and aggregation relationships
- Attributes and attribute-type information
- Methods with parameters
- Navigability
- Dependencies

Additional UML Design and Implementation Diagrams

- **Activity diagrams** model actions that will be performed when an operation is executing as well as the results of those actions.
- **Component diagrams** graphically depict the physical architecture of the software of the system. They can be used to show how programming code is divided into modules and to depict the dependencies between those components.
- **Deployment diagrams** describe the physical architecture of the hardware and software in the system.