Analisis dan Perancangan Berorientasi Objek

**Oleh :** Sintya Sukarta, ST.,MT

(Digunakan di lingkungan sendiri, sebagai buku ajar

mata kuliah Analisis dan Perancangan Berorientasi Objek)

******

**Fakultas Teknik dan Ilmu Komputer**

**Program Studi Manajemen Informatika**

**Universitas Komputer Indonesia**

# Pertemuan 6

## Konsep Dasar UML

Menurut Nugroho (2010:10), Sesungguhnya tidak ada batasan yag tegas diantara berbagai konsep dan konstruksi dalam UML, tetapi untuk menyederhanakannya, kita membagi sejumlah besar konsep dan dalam UML menjadi beberapa *view*. Suatu *view* sendiri pada dasarnya merupakan sejumlah konstruksi pemodelan UML yang merepresentasikan suatu aspek tertentu dari sistem atau perangkat lunak yang sedang kita kembangkan. Pada peringkat paling atas, *view-view* sesungguhnya dapat dibagi menjadi tiga area utama, yaitu: klasifikasi struktural (*structural classification*), perilaku dinamis (*dinamic behaviour*), serta pengolahan atau manajemen model (*model management*).

### Definisi UML

Menurut Widodo, (2011:6), “UML adalah bahasa pemodelan standar yang memiliki sintak dan semantik”.

Menurut Nugroho (2010:6), ”UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek).” Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa UML adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk menvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek (*Object Oriented programming*).

### Diagram

Dalam UML sendiri terdapat beberapa diagram yang wajib dikuasai yaitu:

### Structural Diagram

* Class Diagram, diagram ini terdiri dari class, interface, association, dan collaboration. Diagram ini menggambarkan objek - objek yang ada di sistem.
* Object Diagram, diagram ini menggambarkan hasil instansi dari class diagram. Diagram ini digunakan untuk membuat prototype.
* Component Diagram, diagram ini menggambarkan kumpulan komponen dan hubungan antar komponen. Komponen terdiri dari class, interface, atau collaboration
* Deployment Diagram, diagram ini menggambarkan kumpulan node dan hubungan antar node. Node adalah entitas fisik dimana komponen di-deploy. Entitas fisik ini dapat berupa server atau perangkat keras lainnya.

### Behavioral Diagram

* Use case Diagram, diagram ini menggambarkan kumpulan use case, aktor, dan hubungan mereka. Use case adalah hubungan antara fungsionalitas sistem dengan aktor internal/eksternal dari sistem.
* Sequence Diagram, diagram ini menggambarkan interaksi yang menjelaskan bagaimana pesan mengalir dari objek ke objek lainnya.
* Collaboration Diagram, diagram ini merupakan bentuk lain dari sequence diagram. Diagram ini menggambarkan struktur organisasi dari sistem dengan pesan yang diterima dan dikirim.
* Statechart Diagram, diagram ini menggambarkan bagaimana sistem dapat bereaksi terhadap suatu kejadian dari dalam atau luar. Kejadian (event) ini bertanggung jawab terhadap perubahan keadaan sistem.
* Activity Diagram, menggambarkan aliran kontrol sistem. Diagram ini digunakan untuk melihat bagaimana sistem bekerja ketika dieksekusi.

### View

Use case diagram



Activity Diagram



Sequence diagram



Class diagram



Statemachine diagram



Communication diagram



Deployment diagram



Component diagram



**DAFTAR PUSTAKA**

*Kenneth H. Rossen, Discrete Mathematics and Its Application, 4th edition, 1999, Mc-Graw Hill International*

*Bernard Kolman dan Robert C. Busby, Discrete Mathematical Structures For Computer Science, 2nd edition, 1987, Prentice Hall*

*C. L. Liu, Elements of Discrete Mathematics, 1992, Gramedia Pustaka Utama*

*Rinaldi Munir, Matematika Diskrit, Edisi 2, 2003, Penerbit Informatika Bandung*

*Jong Jek Siang, Matematika Diskrit Dan Aplikasinya Pada Ilmu Komputer, 2002, Penerbit Andi Yogyakarta*