Nama : Isep Lutpi Nur

NPM : 2113191079

MK : Pemodelan Sistem Berbasis Objek

Tugas : Minggu 4 - Pengantar UML

**Unified Modeling Language (UML)**

**Pengertian UML**

UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifacts (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, sperti pada pemodelan bisnis dan non sistem perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object[[1]](#Sumber1).

**Fungsi UML**

Undefined modeling language atau biasa dikenal dengan UML digunakan untuk memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunkan konsep berorientasi object dan juga untuk meciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin[[1]](#Sumber1).

Jadi, UML dibuat untuk memudahkan para system developer untuk berdiskusi dengan bahasa pemodelan yang mudah dipahami.

**Sejarah UML**

UML dimulai secara resmi pada Oktober 1994, ketika Rumbaugh menggabungkan kekuatan dengan Booch. Mereka berdua lalu bekerja bersama di Relational Software Cooperation. Proyek ini memfokuskan pada penyatuan metode booch dan Rumbaugh(OMT). Pada bulan October 1995, UML merilis versi 0.8 dan pada waktu yang sama juga Jacobson bergabung dengan Relational. Cakupan dari UML pun semakin meluas. Kemudian dibangunlah persatuan untuk UML dengan beberapa organisasi yang akan menyumbangkan sumber dayanya untuk bekerja, mengembangkan,dan melengkapi UML[[1]](#Sumber1).

Banyak partner yang berkontribusi pada UML 1.0, diantaranya Digital Equipment Corporation, Hawlett-Packard, I-Logix, IBM, ICON Computing, MCI systemhouse, Microsoft, Oracle, Relation, Texas Insturments dan Unisys. Dari kolaborasi ini dihasilkan UML 1.0 yang merupakan bahasa pemodelan yang ditetapkan secara baik, expressive, kuat dan cocok untuk lingkungan masalah yang luas. Dan pada January 1997, UML dijadikan sebagai standar bahasa pemodelan[[1]](#Sumber1).

**Jenis – Jenis Diagram UML**

1. **Diagram Kelas (Class Diagram):** Kelas Diagram berfungsi untuk menjelaskan tipe dari object sistem dan hubungannya dengan object yang lain. Object adalah nilai tertentu dari setiap attribute kelas entity. Pada penggambaran kelas diagram ada dikenal dengan kelas analisis yaitu kelas ber-stereotype. Tapi yang biasanya dipakai adalah kelas diagram tanpa stereotype[[2]](#Sumber2).
2. **Objek Diagram (Object Diagram):** Object diagram sangat mirip dengan diagram kelas. Perbedaan utama adalah bahwa diagram objek menggambarkan objek dan hubungan mereka. Tujuan utama dari diagram objek adalah untuk memungkinkan analis untuk mengungkap rincian tambahan kelas. Dalam beberapa kasus, pernyataan variabel dari sebuah class diagram dapat membantu pengguna atau analis dalam menemukan atribut tambahan yang relevan, hubungan, dan atau operasi, atau mungkin menemukan bahwa beberapa atribut, hubungan, atau operasi yang salah tempat[[2]](#Sumber2).
3. **Use Case Diagram :** Use case adalah abstraksi dari interaksi antara system dan actor. Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai[[2]](#Sumber2).
4. **Sequence Diagram:** Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya sequence diagram adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan use case diagram. Bersifat dinamis. Diagram urutan adalah interaksi yang menekankan pada pengiriman pesan (message) dalam suatu waktu tertentu[[2]](#Sumber2).
5. **Collaboration Diagram:** Collaboration diagram menjelaskan tentang interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan[[3]](#Sumber3).
6. **Statechart Diagram:** State diagram menggambarkan urutan keadaan yang dilalui objek dalam suatu kelas, karena suatu kejadian menyababkan suatu perpindahan aktivitas/state. State dari objek adalah penggolongan dari satu atau lebih nilai attribute pada kelas[[2]](#Sumber2).
7. **Activity Diagram:** Activity diagram menyediakan analis dengan kemampuan untuk memodelkan proses dalam suatu sistem informasi. Activity diagram dapat digunakan untuk alur kerja model, use case individual, atau logika keputusan yang terkandung dalam metode individual3. Activity diagram juga menyediakan pendekatan untuk proses pemodelan paralel. Activity diagram lebih lanjut[[2]](#Sumber2) .
8. **Component Diagram:** Diagram ini bila dikombinasikan dengan diagram penyebaran dapat digunakan untuk menggambarkan distribusi fisik dari modul perangkat lunak melalui jaringan. Misalnya, ketika merancang sistem client-server, hal ini berguna untuk menunjukkan mana kelas atau paket kelas akan berada pada node klien dan mana yang akan berada di server[[2]](#Sumber2).
9. **Deployment Diagram:** Deployment diagram menggambarkan detail bagaimana komponen di deploy dalam infrastruktur system, dimana komponen akan terletak (pada mesin, server atau piranti keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal. Hubungan antar node ( misalnya TCP/IP) dan requirement dapat juga didefinisikan dalam diagram ini[[2]](#Sumber2).

**Sumber:**

[1]<https://www.codepolitan.com/unified-modeling-language-uml>

[2] <https://www.jejaring.web.id/jenis-jenis-diagram-uml>

[3] Pertemuan 4 - Pengantar UML.pdf (Kuliah online USBYPKP)