OBJECT MODELING TECHNIQUE

Ana Fariza, S.Kom, M.Kom

MATERI

- Pendahuluan
- Rambough OMT
- Booch OMT
- Jacobson OOSE
- Siklus Object-oriented software engineering (OOSE)
- UNIFIED Model Process
- UML View
- Diagram UML

PENDAHULUAN

3

- **object-modeling technique (OMT) adalah bahasa pemodelan obyek untuk pemodelan dan desain perangkat lunak.**
- Dikembangkan oleh Rumbaugh, Blaha, Premerlani, Eddy dan Lorensen sebagai metode untuk pengembangan sistem berorientasi obyek dan mendukung pemrograman berorientasi obyek

COMPANY

RAUMBAUGH OMT

4

- Tujuan pemodelan menuru Raumbaugh (1991)
  - Melakukan testing fisik dari entiti sebelum membangunnya (simulasi)
  - Komunikasi dengan konsumen
  - Visualisasi (alternatif dari presentasi informasi)
  - Mengurangi kompleksitas
- Terdapat 3 jenis model utama
  1. ***Model Obyek*** : konsep utama adalah class dan asosiasi dengan atribut dan operasi. Relasi antar class berupa agregasi dan generalisasi
  2. ***Model Dinamis*** : merepresentasikan state/transisi model. Konsep utama adalah state, transisi antar state dan event yang menyebabkan transisi. Aksi dimodelkan sebagai kejadian dalam state
  3. ***Model Fungsional*** : menangani proses dari model, hubungan ke diagram alir data. Konsep utama adalah proses, data store, data flow dan aktor.

COMPANY

BOOCH OMT

5

- Fase analisa dibagi menjadi beberapa step
  - **Persyaratan Konsumen** : langkah pertama adalah mendapatkan persyaratan dari perspektif konsumen. Menghasilkan deskripsi fungsi dan struktur sistem
  - **Analisa Domain** : dilakukan dengan mendefinisikan class obyek: atribut, pewarisan dan method. Diagram state untuk obyek kemudian dihasilkan.
  - Fase analisa dilengkapi dengan **validation step.**
  - Fase analisa teriterasi antara **persyaratan, analisa domain dan validasi** sampai dicapai konsistensi
  - Setelah fase analisa selesai, metodologi Booch mengembangkan arsitektur dalam fase desain
  - Fase desain berlangsung **iteratif.**
  - Desain logika dipetakan ke desain fisik seperti **proses, performansi, tipe data, struktur data, visibilitas**
  - Prototipe dibuat dan di test. Proses iterasi terjadi antara **desain logika, desain fisik, prototipe dan testing**
  - Metodologi Booch terjadi secara terurut, fase analisa selesai dan dilanjutkan fase desain diselesaikan
  - Siklus pada setiap fase dibagi menjadi siklus yang lebih kecil.

COMPANY

JACOBSON OOSE

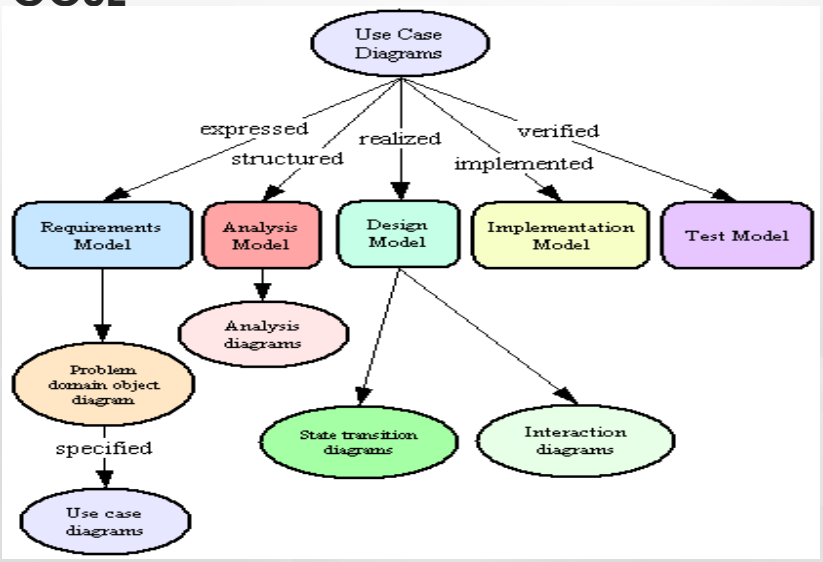
6

- Object-Oriented Software Engineering (OOSE) teknik desain perangkat lunak yang digunakan dalam pemrograman berorientasi obyek.
- OOSE dikembangkan oleh **Ivar Jacobson** tahun **1992.**
- OOSE adalah metodologi desain berorientasi obyek yang menggunakan use case dalam desain perangkat lunak.
- Termasuk di dalam OOSE: model persyaratan (requirement), analisis, desain, implementasi dan testing.

COMPANY

OOSE

7



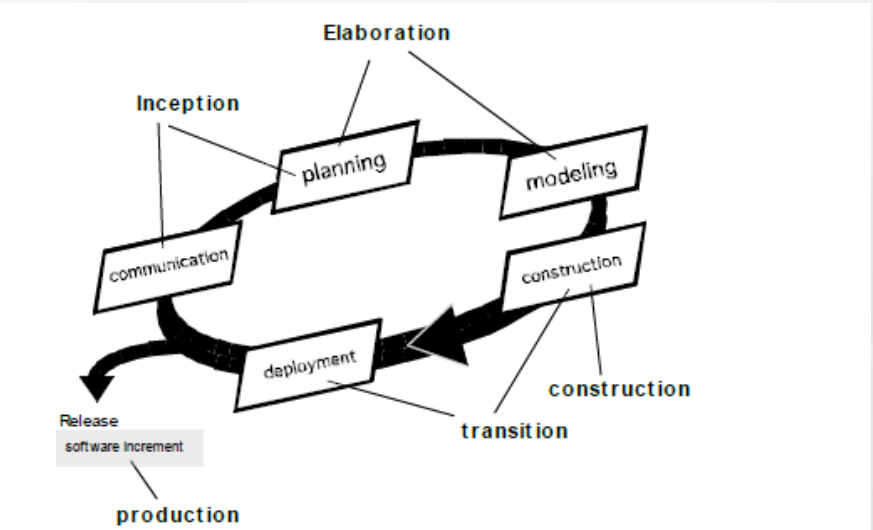
SIKLUS OOSE

8

- **Requirements Engineering (Rekayasa Persyaratan) :**
  - Memperoleh persyaratan (requirement)
  - Analisa berorientasi obyek
  - spesifikasi
- **Desain OO:**
  - Desain arsitektur
  - Desain detail
  - Keduanya membutuhkan transformasi dari persyaratan fungsional menjadi elemen desain OO
- **Implementasi atau Coding:**
  - Pemrograman menggunakan bahasa pemrograman OO dan tool.
- **Testing:**
  - Testing unit: metode tes dalam setiap obyek
  - Testing integrasi: kolaborasi tes antar obyek
  - Testing sistem: tes keseluruhan sistem sebagai kumpulan obyek
  - Testing penerimaan (acceptance testing): tes untuk standarisasi dan kepuasan konsumen.

UNIFIED MODEL PROCESS

9



UNIFIED MODEL PROCESS

10

- **Iteratif dan Incremental:**
  - Proses Unified adalah proses pengembangan iteratif dan incremental.
  - Fase elaborasi (perluasan), konstruksi dan transisi dibagi ke dalam iterasi waktu.
  - Setiap iterasi menghasilkan *increment*, yang menghasilkan fungsi yang lebih baik/lengkap.
- **Use Case Driven:**
  - Dalam proses Unified, use case digunakan untuk menangkap persyaratan fungsional dan menentukan isi dari iterasi.
- **Fokus pada Resiko:**
  - Proses Unified membutuhkan tim proyek untuk fokus pada resiko yang kiritis dalam siklus proyek.
  - Penyampaian (delivery) setiap iterasi, terutama pada fase elaborasi, harus dipilih untuk menjamin resiko yang sangat besar terjadi di awal.

PENGENALAN UML

11

- UML adalah **bahasa yang digunakan untuk analisa dan desain berorientasi obyek.**
- UML terdiri dari kumpulan notasi grafis untuk membuat model visual dari sistem perangkat lunak.
- Grady Booch, Ivar Jacobson dan James Rumbaugh adalah 3 pengembang UML.
- UML adalah bahasa untuk visualisasi, spesifikasi, konstruksi dan dokumentasi.
- UML **bukan metode pengembangan**. UML didesain cocok dengan metode pengembangan berorientasi obyek.

COMPANY

VIEW PADA UML

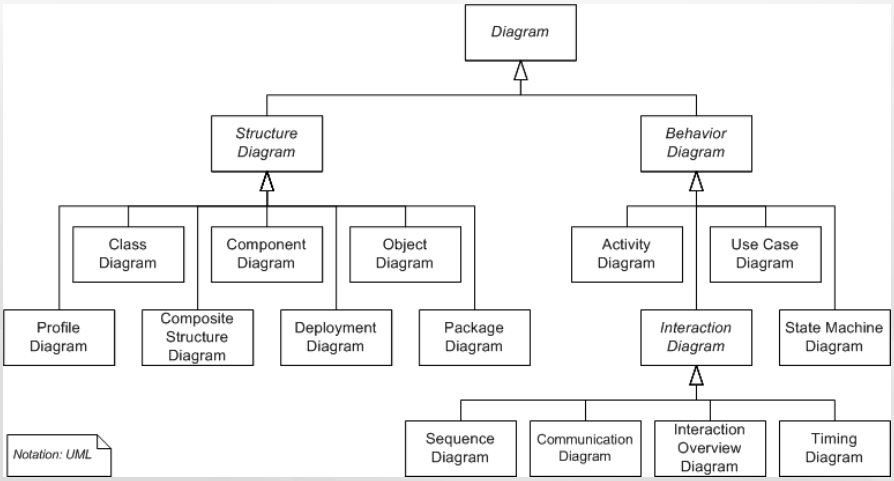
12

1. **Functional View:**
   - Menggambarkan **persyaratan fungsional** dari sistem.
   - **Diagram use case** merupakan **static functional view** untuk fungsi dan relasi statis.
   - **Diagram aktifitas** merupakan **dynamic functional view.**
2. **Static Structural View:**
   - **Diagram class** dan **diagram obyek** merupakan **structural view** dari sistem.
3. **Behavioral (dynamic structural) view:**
   - **Diagram interaktif** dan **diagram collaboration** menggambarkan **urutan interaksi antar obyek.**
   - **Diagram state** menunjukkan **perilaku obyek berbasis state.**
4. **Architectural View:**
   - Menggambarkan struktur logika dan fisik.
   - Menggunakan **diagram komponen** dan **diagram deployment**.

COMPANY

VIEW PADA UML

13



COMPANY

DIAGRAM UML

14

- Diagram adalah kunci dari UML. Diagram ini dikategorikan menjadi 2 yaitu:
  - **Diagram struktural** terdiri dari diagram statis seperti diagram class, diagram obyek dll.
  - **Diagram perilaku** terdiri dari diagram dinamis seperti diagram sequence, diagram collaboration dll.
- Perilaku statis dan dinamis dari sistem divisualisasikan menggunakan diagram-diagram ini.

COMPANY

DIAGRAM CLASS

15

- Diagram Class adalah diagram yang paling popular dalam UML yang digunakan oleh komunitas berorientasi obyek.
- Menggambarkan obyek dalam sistem dan relasinya.
- Diagram class terdiri dari atribut dan fungsi.
- Satu diagram class menggambarkan aspek khusus dari sistem dan kumpulan diagram class menggambarkan keseluruhan sistem.
- Diagram class menggambarkan *static view* dari sistem.
- Hanya diagram class yang dapat dipetakan menjadi bahasa berorientasi obyek.
- Digunakan secara luas oleh developer.

COMPANY

DIAGRAM OBYEK

16

- Diagram obyek adalah anggota dari diagram class.
- Elemen dasar sama dengan diagram class.
- Diagram obyek terdiri dari obyek dan hubungan antar obyek.
- Menangkap anggota dari sistem pada waktu tertentu.
- Diagram obyek digunakan untuk prototipe, reverse engineering dan memodelkan skenario praktek.

COMPANY

DIAGRAM KOMPONEN

17

- Diagram komponen adalah diagram UML khusus untuk menggambarkan implementasi statis dalam sistem.
- Diagram komponen terdiri dari komponen fisik seperti library, file, folder dll.
- Diagram ini digunakan untuk perspektif implementasi.
- Lebih dari satu diagram komponen digunakan untuk menggambarkan keseluruhan sistem.
- Teknik forward dan reverse engineering digunakan untuk menjadikan sistem yang dapat dieksekusi dari diagram komponen.

COMPANY

DIAGRAM DEPLOYMENT

18

- Diagram komponen digunakan untuk menggambarkan static view dari deployment sistem.
- Diagram ini utamanya digunakan oleh system engineer.
- Diagram deployment terdiri dari dari node (simpul) dan relasi antar simpul.
- Diagram deployment yang efisien adalah bagian integral dari pengembangan aplikasi perangkat lunak.

COMPANY

DIAGRAM USE CASE

19

- Diagram use case digunakan untuk menangkap perilaku dinamis dari sistem.
- Terdiri dari use case, aktor dan relasi nya.
- Diagram use case digunaan pada desain level tinggi untuk menangkap persyaratan sistem.
- Menggambarkan fungsionalitas sistem dan alirannya.
- Meskipun diagram use case bukan kandidat yang baik untuk forward dan reverse engineering, tetapi tetap digunakan untuk memodelkan dengan cara yang berbeda.

COMPANY

DIAGRAM INTERAKSI

20

- Diagram interaksi digunakan untuk menangkap perilaku dinamis pada sistem.
- Diagram sequence dan diagram collaboration adalah diagram interaksi yang digunakan untuk tujuan tsb.
- Diagram sequence menjelaskan urutan waktu dari aliran pesan, digram collaboration digunakan untuk mengerti organisasi terstruktur dari sistem.
- Umumnya diagram sequence dan collaboration digunakan untuk menggambarkan keseluruhan sistem.

COMPANY

DIAGRAM STATECHART

21

- Diagram statechart adalah satu dari 5 diagram yang digunakan untuk memodelkan perilaku dinamis dari sistem.
- Diagram ini digunakan untuk memodelkan keseluruhan siklus dari obyek.
- Diagram aktifitas adalah salah satu bentuk diagram statechart.
- State dari obyek didefinisikan sebagai kondisi dimana obyek berada dalam waktu tertentu dan obyek berpindah ke state lain jika terjadi event.
- Diagram statechart juga digunakan untuk forward dan reverse engineering.

COMPANY

DIAGRAM AKTIFITAS

22

- Diagram aktifitas adalah diagram yang penting untuk menggambarkan perilaku dinamis.
- Diagram aktifitas terdiri dari aktifitas, link, hubungan dll.
- Memodelkan semua jenis aliran baik paralel, single, konkuren dll.
- Diagram aktifitas menggambarkan kontrol aliran dari satu aktifitas ke aktifitas lain tanpa pesan tertentu.
- Diagram ini digunakan untuk memodelkan pada level tinggi dari persyaratan bisnis.

COMPANY