# Analysis Modeling

Ir. I Gede Made Karma, MT

## Analysis Modeling

- Data modeling
- Functional modeling and information flow
- Behavioral modeling
- Structured analysis

## Overview

- The analysis model is the first technical representation of a system.
- Analysis modeling uses a combination of text and diagrams to represent software requirements (data, function, and behavior) in an understandable way.
- Building analysis models helps make it easier to uncover requirement inconsistencies and omissions.
- Two types of analysis modeling are commonly used:
  - structured analysis (discussed in this chapter) and
  - object-oriented analysis (discussed in Chapter 21).
- Data modeling uses entity-relationship diagrams to define data objects, attributes, and relationships.
- Functional modeling uses data flow diagrams to show how data are transformed inside the system.
- Behavioral modeling uses state transition diagrams to show the impact of events.
- Analysis work products must be reviewed for completeness, correctness, and consistency.
- The SEPA web site contains descriptions of several classical analysis techniques (DSSD, JSD, SADT).

## Structured Analysis (DeMarco)

- Analysis products must be highly maintainable, especially the software requirements specification.
- Problems of size must be dealt with using an effective method of partitioning.
- Graphics should be used whenever possible.
- Differentiate between the logical (essential) and physical (implementation) considerations.
- Find something to help with requirements partitioning and document the partitioning before specification.
- Devise a way to track and evaluate user interfaces.
- Devise tools that describe logic and policy better than narrative text.

## Analysis Model Objectives

- Describe what the customer requires.
- Establish a basis for the creation of a software design.
- Devise a set of requirements that can be validated once the software is built.

## Analysis Model Elements

- Data dictionary - contains the descriptions of all data objects consumed or produced by the software
- Entity relationship diagram (ERD) - depicts relationships between data objects
- Data flow diagram (DFD) - provides an indication of how data are transformed as they move through the system; also depicts functions that transform the data flow (a function is represented in a DFD using a process specification or PSPEC)
- State transition diagram (STD) - indicates how the system behaves as a consequence of external events, states are used to represent behavior modes. Arcs are labeled with the events triggering the transitions from one state to another (control information is contained in control specification or CSPEC)

## Data Modeling Elements (ERD)

- Data object - any person, organization, device, or software product that produces or consumes information
- Attributes - name a data object instance, describe its characteristics, or make reference to another data object
- Relationships - indicate the manner in which data objects are connected to one another

## Cardinality and Modality (ERD)

- Cardinality - in data modeling, cardinality specifies how the number of occurrences of one object are related to the number of occurrences of another object (1:1, 1:N, M:N)
- Modality - zero (0) for an optional object relationship and one (1) for a mandatory relationship

## Functional Modeling and Information Flow (DFD)

- Shows the relationships of external entities, process or transforms, data items, and data stores
- DFD's cannot show procedural detail (e.g. conditionals or loops) only the flow of data through the software
- Refinement from one DFD level to the next should follow approximately a 1:5 ratio (this ratio will reduce as the refinement proceeds)
- To model real-time systems, structured analysis notation must be available for time continuous data and event processing (e.g. Ward and Mellor or Hately and Pirbhai)

## Behavioral Modeling (STD)

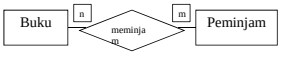
- State transition diagrams represent the system states and events that trigger state transitions
- STD's indicate actions (e.g. process activation) taken as a consequence of a particular event
- A state is any observable mode of behavior
- Hatley and Pirbhai control flow diagrams (CFD) can also be used for behavioral modeling

## Creating Entity Relationship Diagrams

- Customer asked to list "things" that application addresses, these things evolve into input objects, output objects, and external entities
- Analyst and customer define connections between the objects
- One or more object-relationship pairs is created for each connection
- The cardinality and modality are determined for an object-relationship pair
- Attributes of each entity are defined
- The entity diagram is reviewed and refined

## E-R Diagram (1)

- Contoh:




- Entitas:
  - Buku
    - Atribut: ISBN, Judul, Pengarang, Penerbit, ...
  - Peminjam
    - Atribut: NIM, Nama, Alamat, ...

## ER Diagram (2)

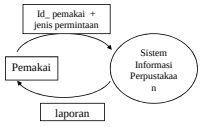
- Relasi:
  - Meminjam
    - Atribut: ISBN, NIM, ...
- Kardinalitas:
  - N-M
    - 1 buku dapat dipinjam oleh banyak peminjam dan
    - 1 peminjam dapat meminjam banyak buku
- Catatan:
  - bedakan ERD dalam level abstraksi permasalahan sistem dengan ERD dalam level abstraksi kebutuhan PL

## Creating Data Flow Diagram

- Level o data flow diagram should depict the system as a single bubble
- Primary input and output should be carefully noted
- Refinement should begin by consolidating candidate processes, data objects, and data stores to be represented at the next level
- Label all arrows with meaningful names
- Information flow must be maintained from one level to level
- Refine one bubble at a time
- Write a PSPEC (a "mini-spec" written using English or another natural language or a program design language) for each bubble in the final DFD

## Context Diagram

- Merepresentasikan sistem sebagai sebuah 'black box' terhadap lingkungan sekitarnya
- Contoh:




## Data Flow Diagram (1)

- Penjabaran lebih lanjut dari Diagram Konteks
- dapat terdiri atas beberapa level
  - level 0: level tertinggi
  - level 1: penjabaran dari level 0
  - level 2: penjabaran dari level 1, dst
- semakin rendah levelnya, semakin rinci fungsinya
- Catatan:
  - bedakan DFD dalam level abstraksi permasalahan sistem dengan DFD dalam level abstraksi kebutuhan PL

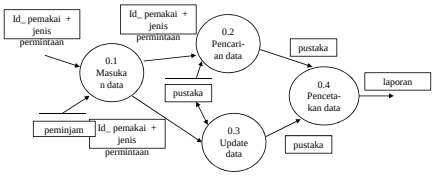
## Data Flow Diagram (2)

- Notasi dasar:


- Setiap proses harus diberi nomor:
  - level.nomor-urut

## Data Flow Diagram (3)

- Contoh level 0:



## Process Specification (1)

- Deskripsi rinci setiap proses yang muncul pada DFD
  - proses yang harus mengandung P-SPEC adalah proses yang sudah tidak didekomposisi lagi menjadi sub-proses dibawahnya (sudah level terendah)

## Process Specification (2)

- Contoh:
  - P-SPEC 0.4:
    - Input:
      - id_pemakai
      - data buku
    - Output:
      - file teks
    - Algoritma:

```
if found then
    print header
else . . .
```

## Creating Control Flow Diagrams

- Creating Control Flow Diagrams
- Begin by stripping all the data flow arrows form the DFD
- Events (solid arrows) and control items (dashed arrows) are added to the diagram
- Add a window to the CSPEC (contains an STD that is a sequential specification of the behavior) for each bubble in the final CFD

## Data Dictionary Contents

- Data Dictionary Contents
- Name - primary name for each data or control item, data store, or external entity
- Alias - alternate names for each data object
- Where-used/how-used - a listing of processes that use the data or control item and how it is used (e.g. input to process, output from process, as a store, as an external entity)
- Content description - notation for representing content
- Supplementary information - other data type information, preset values, restrictions, limitations, etc.

## Data Dictionary (1)

- Menyimpan semua objek data yang dibutuhkan dan dihasilkan oleh PL
  - objek data yang muncul pada:
    - ERD
    - DFD
    - STD
  - harus selengkap dan serinci mungkin
    - contoh: Nama = nama_depan + nama_belakang

## Data Dictionary (2)

- Berisi:
  - Name
    - nama utama yang muncul pada objek data, data store, atau external entity
  - Alias
    - nama lain yang digunakan
  - Where-used/how-used
    - daftar proses yang menggunakan data dan bagaimana menggunakannya
  - Content description
    - notasi untuk merepresentasikan isi data
  - Supplementary information

## Data Dictionary (3)

- Notasi:

| Jenis | Notasi | Arti |
|---|---|---|
| | = | Terdiri atas |
| urutan | + | dan |
| pilihan | [ \| ] | atau |
| pengulangan | { } $^n$ | Pengulangan sebanyak n kali |
| | ( ) | Data optional |
| | * * | pembatas komentar |

## Data Dictionary (4)

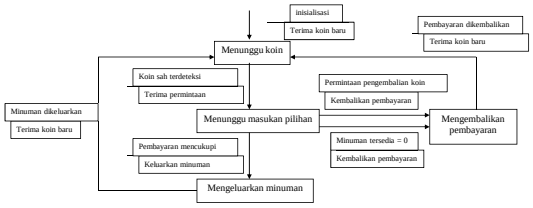
- Contoh:
  - nama mahasiswa = nama depan + nama belakang
  - jenis kelamin = [perempuan | laki-laki]
  - nomor telepon = (kode negara) + kode wilayah + nomor

## Behavioral Modeling

- Mendeskripsikan status sistem yang dapat muncul ketika perangkat lunak digunakan
- mendeskripsikan kelakuan sistem
- Tools:
  - State Transition Diagram
  - Control Specification
- Umumnya digunakan pada sistem waktu-nyata

## State Transition Diagram

- Contoh STD untuk mesin otomatis penjual minuman (tidak ada hubungannya dengan contoh sebelumnya):




## Control Specification

- Fungsi C-SPEC sama dengan P-SPEC namun berisi deskripsi dari setiap status yang dapat muncul pada sistem

## Kaitan antara Data dan Control Model