

Software Engineering: A Practitioner's Approach, 6/e

Chapter 8

Pemodelan Analisis

copyright © 1996, 2001, 2005

R.S. Pressman & Associates, Inc.

A. Haidar Mirza, M.Kom

Siti Sa'uda, M.Kom.

For University Use Only

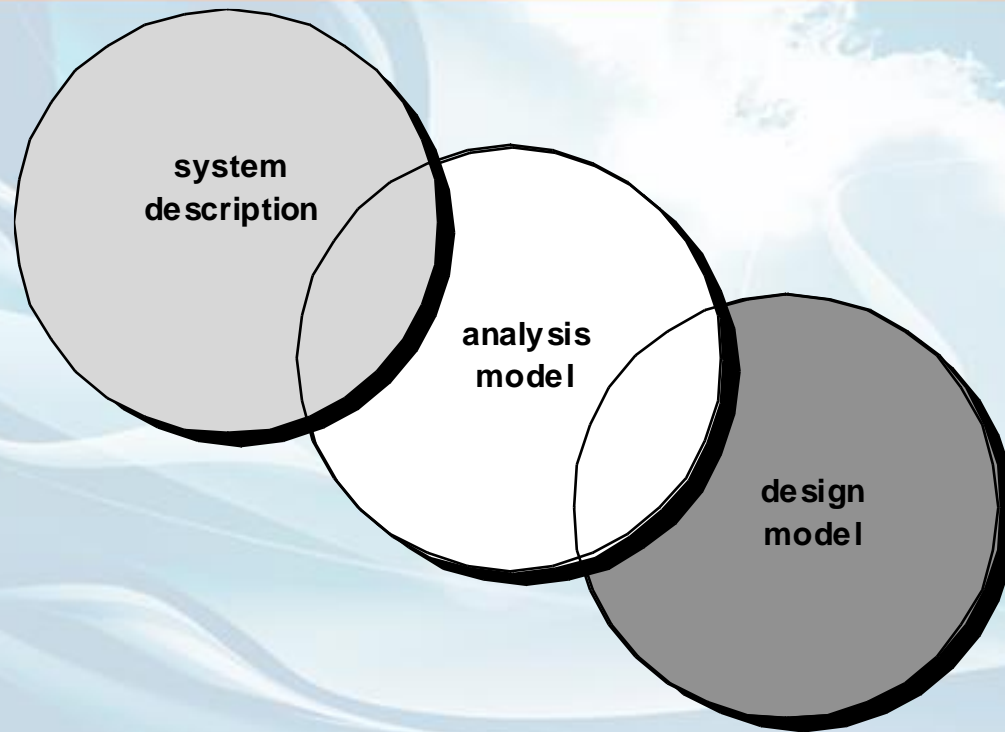
May be reproduced ONLY for student use at the university level
when used in conjunction with *Software Engineering: A Practitioner's Approach*.

Any other reproduction or use is expressly prohibited.

Analisis Kebutuhan

- Analisis Kebutuhan
 - Menentukan karakteristik operasional PL
 - Menunjukkan antarmuka PL dengan elemen sistem yang lain
 - Membuat batasan yang harus dipenuhi PL
- Analisis Kebutuhan memungkinkan Software Engineer (disebut analis atau modeler) untuk :
 - Memperinci kebutuhan dasar yang dibuat kapda rekayasa kebutuhan sebelumnya
 - Membangun model yang dapat menggambarkan skenario user, aktivitas fungsional, class masalah dan relasinya, sistem dan perilaku class, dan aliran data ketika ditransformasikan.

Sebuah Jembatan



Aturan-Aturan

- Model harus fokus pada masalah atau domain bisnis. Tingkat abstraksinya relatif harus lebih tinggi.
- Setiap elemen model analisis sebaiknya memberikan tambahan pada pemahaman keseluruhan kebutuhan PL dan menyediakan wawasan pada domain informasi, fungsi dan perilaku sistem.
- Tunda semua pertimbangan infrastruktur dan model non fungsional hingga fase desain.
- Minimalisasi rangkaian melalui sistem.
- Pastikan model analisis menyediakan nilai untuk semua stakeholder.
- Jaga model sesederhana mungkin.

Analisis Domain

Analisis domain PL adalah identifikasi, analisis, dan spesifikasi kebutuhan umum dari domain aplikasi tertentu, yang biasanya digunakan kembali pada project lain di dalam domain aplikasi yang sama [Analisis domain berorientasi objek adalah] identifikasi, analisis dan spesifikasi kemampuan umum, kemampuan digunakan kembali dalam domain tertentu dalam istilah-istilah objek, class, subassemblies dan framework umum

Donald Firesmith

Analisis Domain

- Tentukan domain yang ingin diinvestigasi.
- Kumpulkan contoh representatif aplikasi pada domain tersebut.
- Analisis setiap aplikasi pada contoh.
- Kembangkan model analisis untuk objek.

Pemodelan Data

- Memeriksa objek data secara independen terhadap proses
- Fokus perhatian pada domain data
- Membuat sebuah model pada abstraksi level konsumen
- Mengindikasikan bagaimana objek data berhubungan satu dengan yang lain

What is a Data Object?

Object —something that is described by a set of attributes (data items) and that will be manipulated within the software (system)

- each instance of an object (e.g., a book) can be identified uniquely (e.g., ISBN #)
- each plays a necessary role in the system i.e., the system could not function without access to instances of the object
- each is described by attributes that are themselves data items

Objek-Objek Umum

- *Entitas eksternal* (printer, user, sensor)
- *Sesuatu* (laporan, display, sinyal)
- *Kejadian atau event* (interupsi, alarm)
- *orang* (manager, engineer, salesperson)
- *Unit organisasi* (divisi, tim)
- *tempat* (rantai pabrik)
- *struktur* (employee record)

Objek Data dan Atribut

Sebuah objek data terdiri dari sekumpulan atribut yang bertindak sebagai aspek, kualitas, karakteristik, atau penjelas objek

object: automobile

attributes:

make

model

body type

price

options code

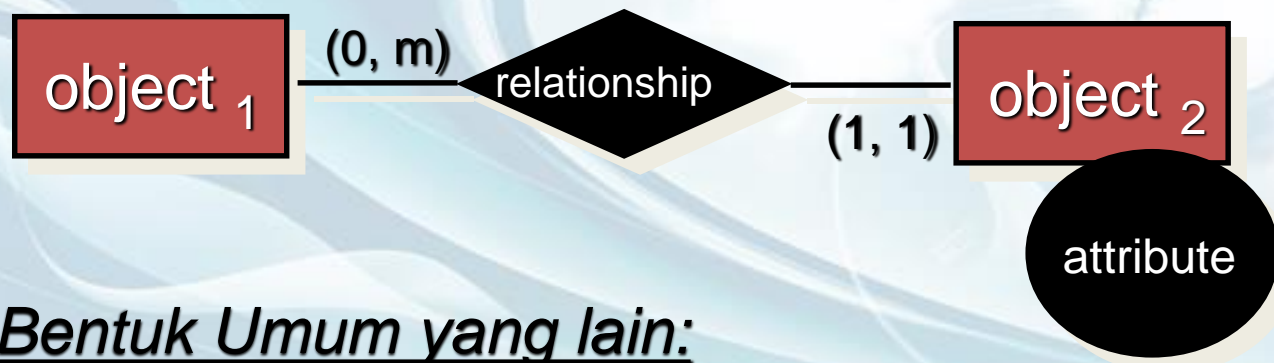
Apakah Relationship?

Relationship – menandakan kaitan, sebuah fakta yang harus diingat oleh sistem, tidak dikomputasi atau diturunkan secara mekanis

- several instances of a relationship can exist
- objects can be related in many different ways

Notasi ERD

Satu bentuk umum:



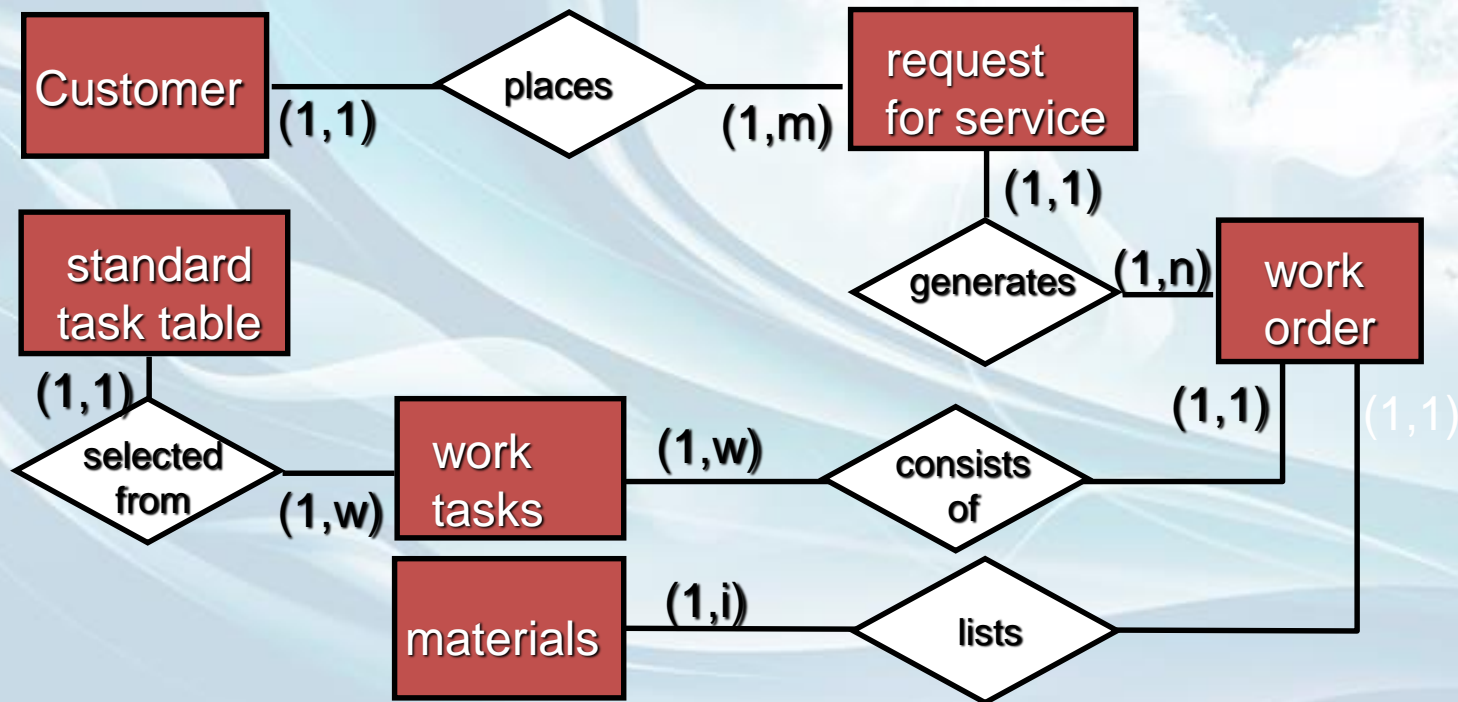
Bentuk Umum yang lain:



Membangun Sebuah ERD

- Level 1—modelkan semua objek data (entitas) dan koneksinya dengan yang lain
- Level 2—modelkan semua entitas dan relasi
- Level 3—modelkan semua entitas, relasi, dan atribut yang menyediakan informasi yang lebih mendalam

ERD: sebuah contoh



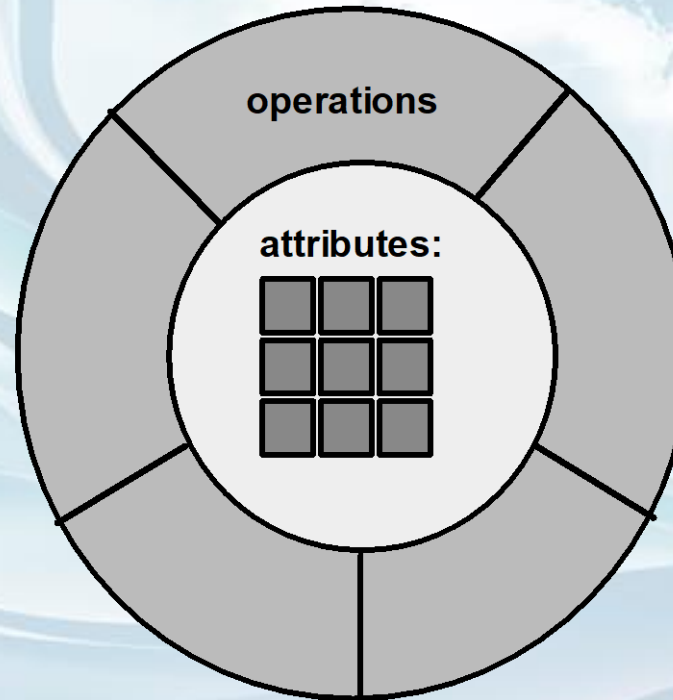
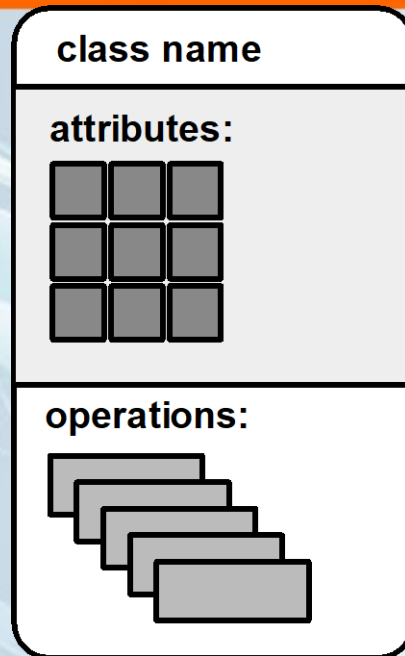
Konsep Object-Oriented

- Harus dipahami untuk menerapkan elemen berbasis class pada model analisis
- Konsep-konsep kunci:
 - Classes dan objects
 - Attributes dan operations
 - Encapsulation dan instantiation
 - Inheritance

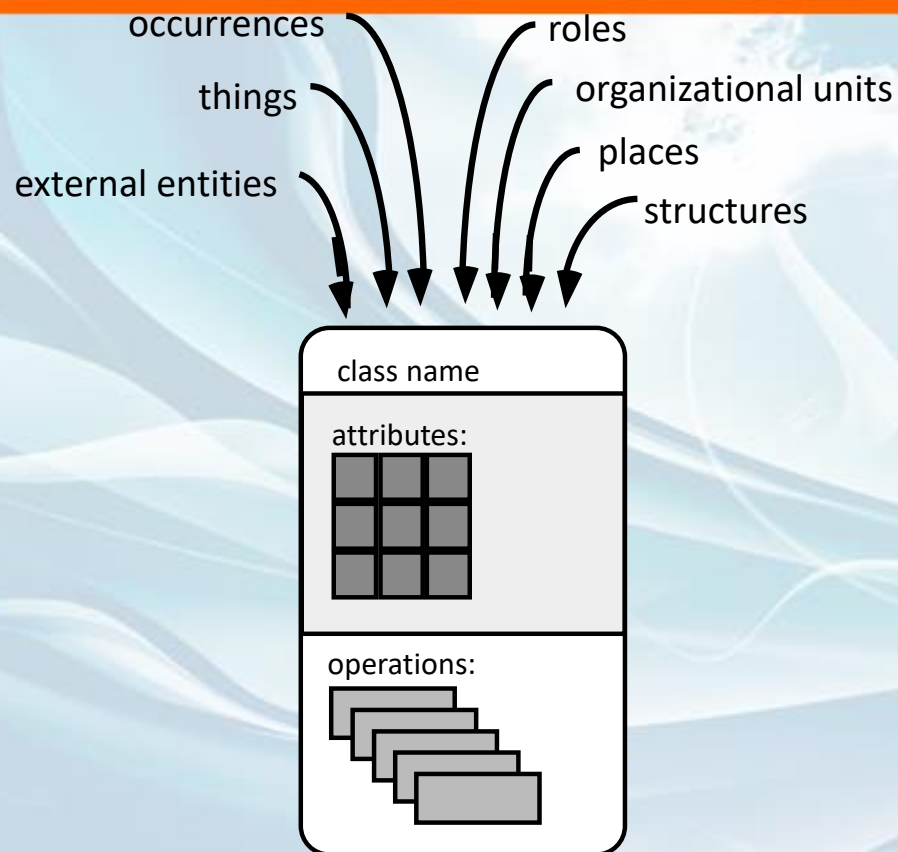
Class

- Pemikiran object-oriented dimulai dengan sebuah class, sering didefinisi sebagai :
 - template
 - deskripsi umum
 - “blueprint” ... Menggambarkan sekelompok item yang mirip
- sebuah **metaclass** (sering disebut **superclass**) yang membangun hierarki semua class yang ada
- Sekali sebuah class item ditentukan, instance spesifik dari class tersebut dapat diidentifikasi

Membangun Class

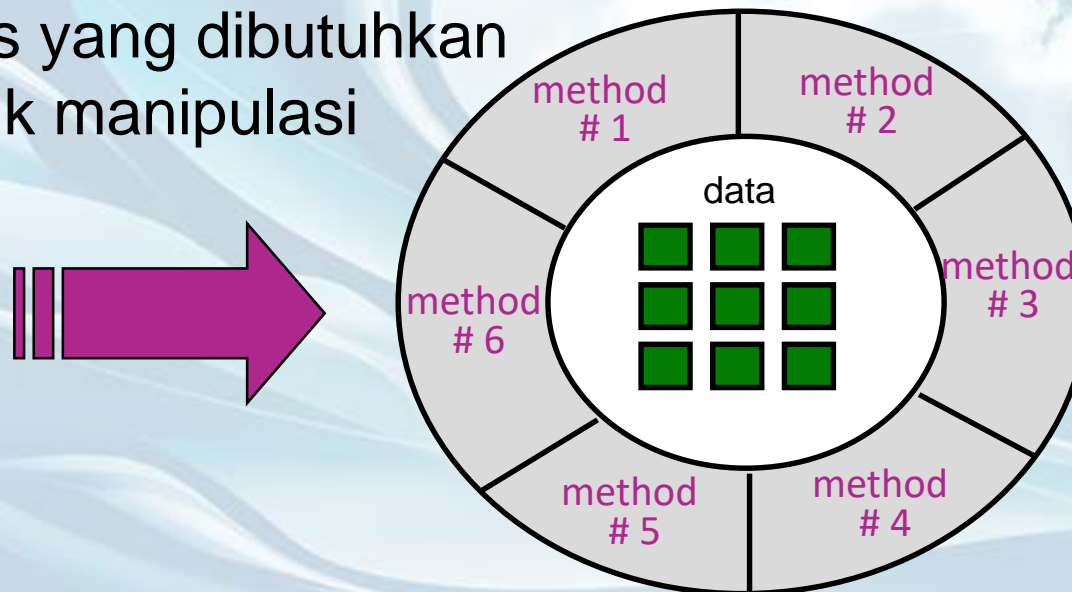


Apakah Class?



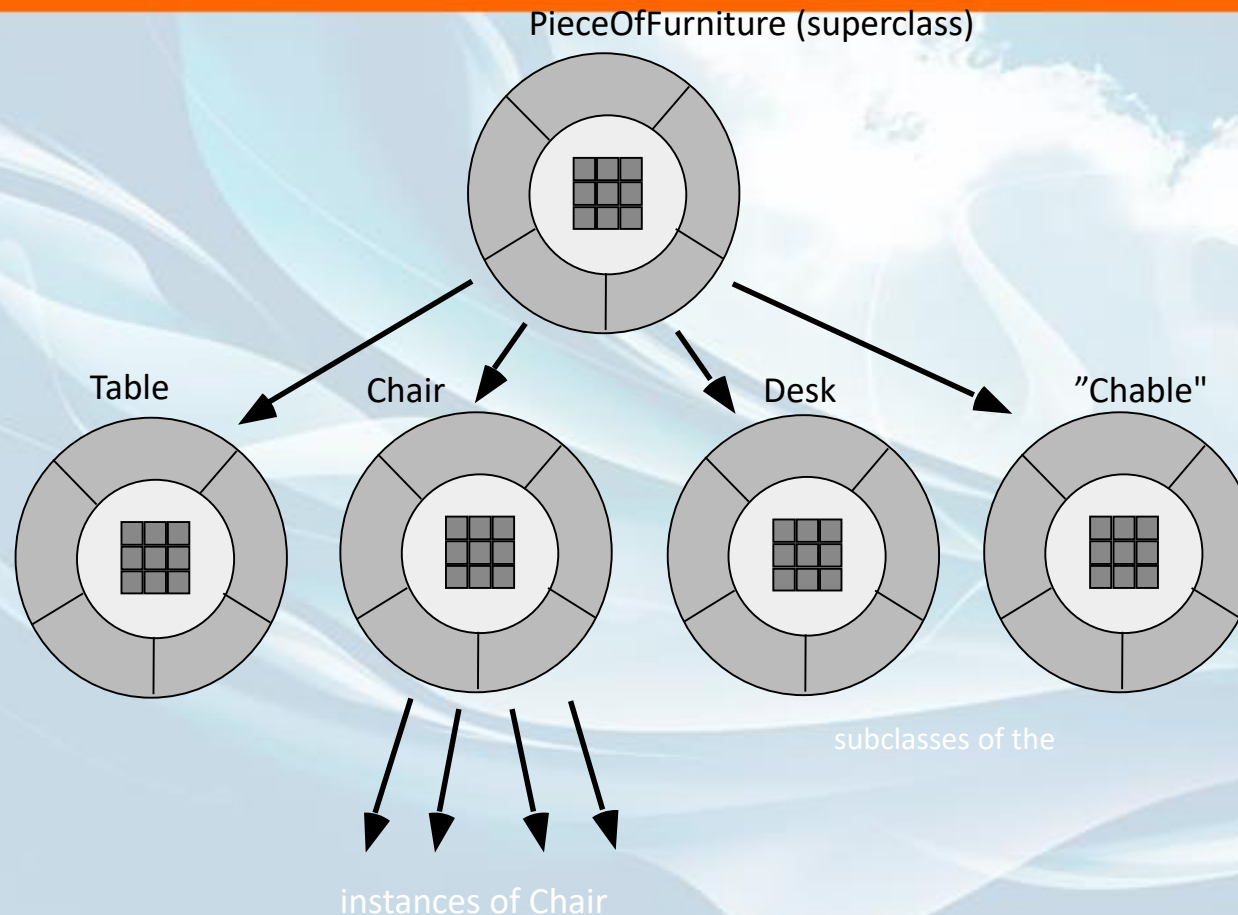
Enkapsulasi/Penyembunyian

Objek mengenkapsulasi
Baik data dan prosedur
Logis yang dibutuhkan
Untuk manipulasi
data



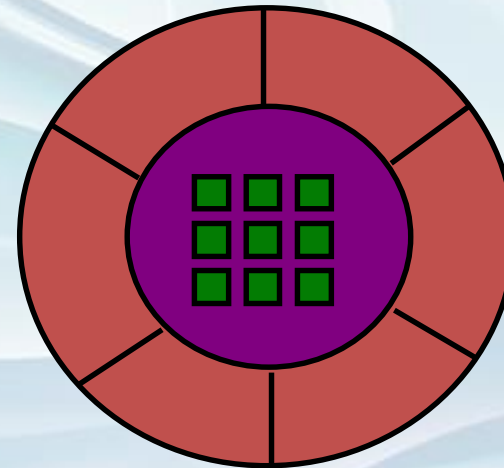
Achieves “information hiding”

Hierarki Class



Method (Operasi, Layanan)

Prosedur yang terenkapsulasi pada sebuah class dan didesain untuk beroperasi pada satu atau lebih atribut data yang ditentukan sebagai bagian dari class. Method dipanggil melalui pesan



Model berbasis Scenario

“[Use-cases] adalah bantuan untuk mendefinisikan apa yang ada pada sistem (aktor) dan apa yang harus dilakukan sistem (use-cases).” Ivar Jacobson

- (1) Apa yang harus ditulis?
- (2) Berapa banyak kita harus menulisnya?
- (3) Sedetail apa gambaran kita ?
- (4) Bagaimana kita mengatur deskripsi?

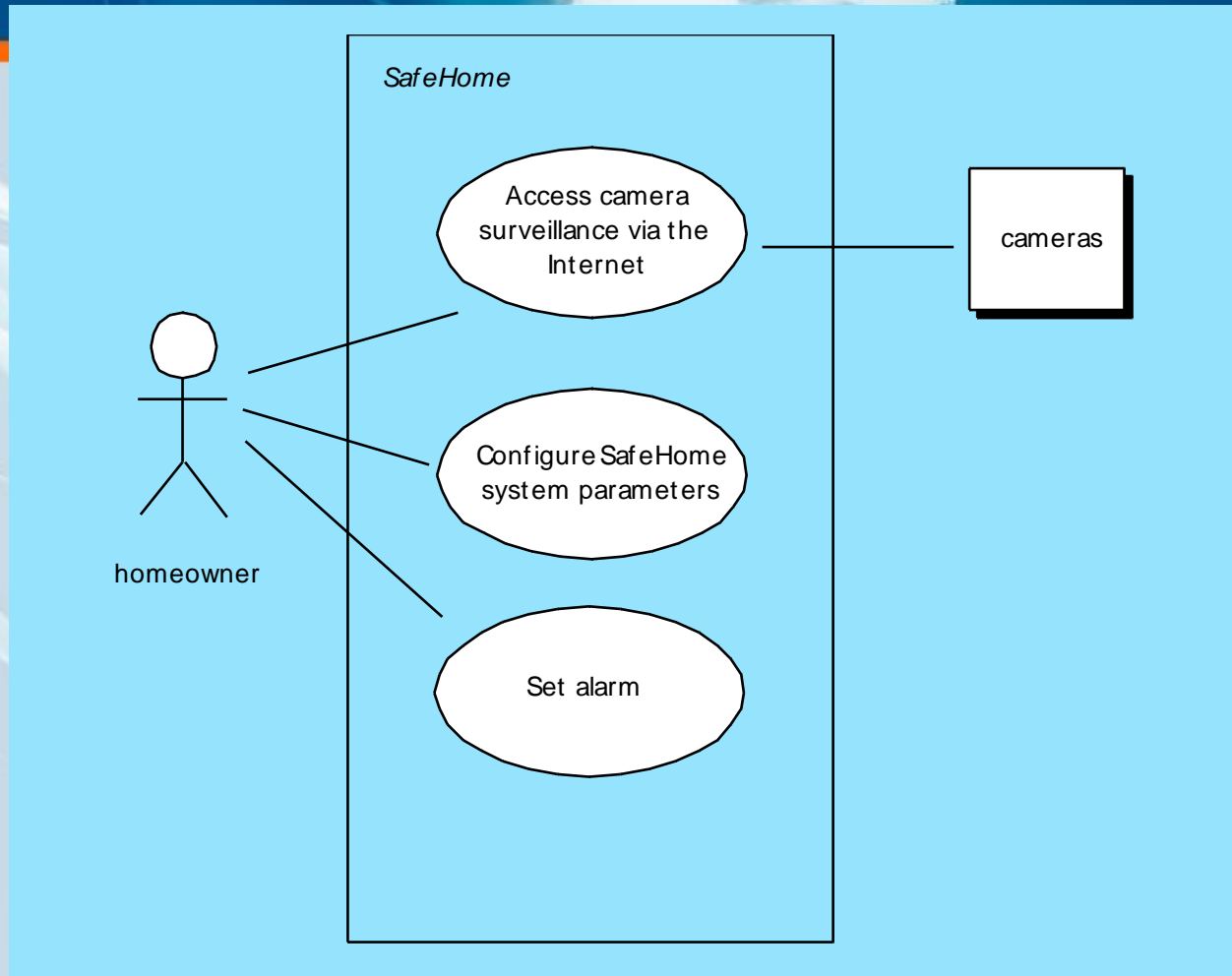
Use-Cases

- Sebuah skenario yang menggambarkan rangkaian kegunaan pada sistem
- *actors* mewakili peran orang atau piranti yang dimainkan ketika sistem berfungsi
- *users* dapat berperan sebagai lebih dari satu peran dalam sebuah skenario yang ditentukan

Mengembangkan Use-Case

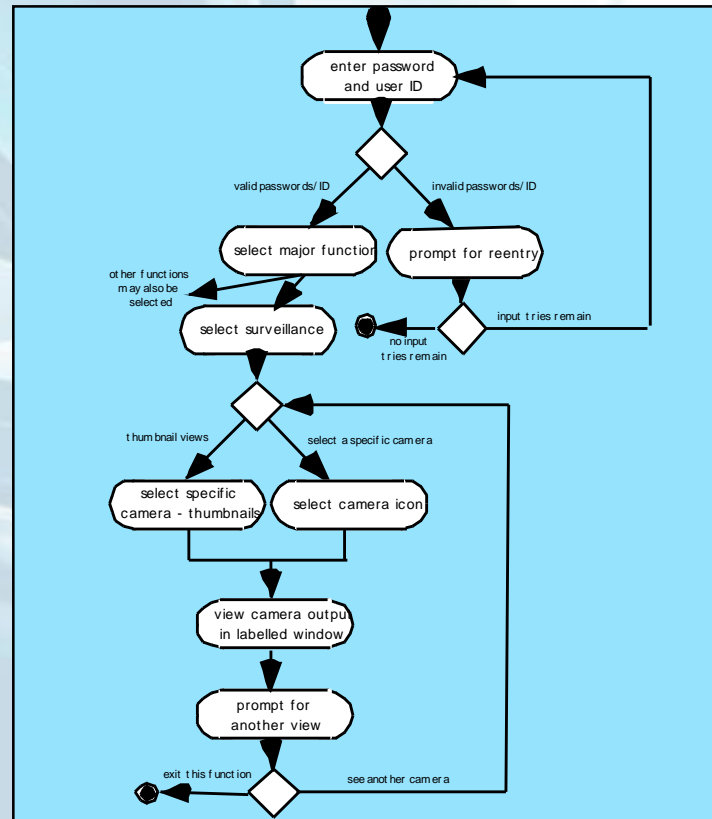
- Apa tugas atau fungsi utama yang harus dilakukan aktor ?
- Sistem Informasi seperti apa yang diperlukan, dihasilkan atau diubah oleh aktor ?
- Apakah aktor harus menginformasikan sistem tentang perubahan dalam lingkungan eksternal?
- Informasi apa yang diharapkan aktor dari sistem?
- Apakah aktor menginginkan diberitahu tentang perubahan yang tidak terduga?

Use-Case Diagram



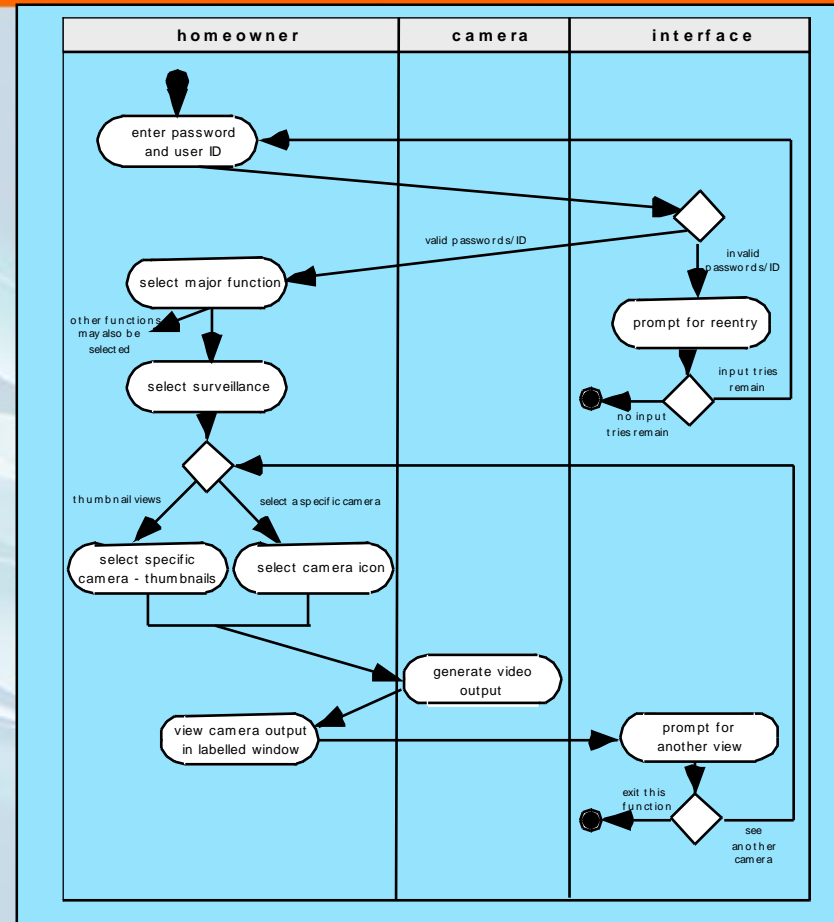
Activity Diagram

Melengkapi use-case dengan menyediakan representasi diagram dari aliran prosedural.



Swimlane Diagrams

Memungkinkan untuk menampilkan aliran aktivitas yang digambarkan oleh use-case, dan di saat yang sama mengindikasikan aktor yang mana, atau class analisis yang mempunyai tanggungjawab terhadap tindakan yang digambarkan oleh kotak aktivitas



Tugas

- Buatlah sebuah paper yang menjelaskan tentang UML beserta contoh kasusnya, minimal 15 halaman, spasi 1.5, kertas A4.
- Kriteria penilaian :
 - Validitas/kebenaran
 - Kelengkapan
 - Kompleksitas kasus

Panduan Spesifikasi

- ❑ use a layered format that provides increasing detail as the "layers" deepen
- ❑ use consistent graphical notation and apply textual terms consistently (stay away from aliases)
- ❑ be sure to define all acronyms
- ❑ be sure to include a table of contents; ideally, include an index and/or a glossary
- ❑ write in a simple, unambiguous style (see "editing suggestions" on the following pages)
- ❑ always put yourself in the reader's position, "Would I be able to understand this if I wasn't intimately familiar with the system?"

Panduan Spesifikasi

Be on the lookout for persuasive connectors, ask why?

keys: *certainly, therefore, clearly, obviously, it follows that ...*

Watch out for vague terms

keys: *some, sometimes, often, usually, ordinarily, most, mostly ...*

When lists are given, but not completed, be sure all items are understood

keys: *etc., and so forth, and so on, such as*

Be sure stated ranges don't contain unstated assumptions

e.g., *Valid codes range from 10 to 100.* Integer? Real? Hex?

Beware of vague verbs such as *handled, rejected, processed, ...*

Beware "passive voice" statements

e.g., *The parameters are initialized.* By what?

Beware "dangling" pronouns

e.g., *The I/O module communicated with the data validation module and its control flag is set.* Whose control flag?

Panduan Spesifikasi

When a term is explicitly defined in one place, try substituting the definition for other occurrences of the term

When a structure is described in words, draw a picture

When a structure is described with a picture, try to redraw the picture to emphasize different elements of the structure

When symbolic equations are used, try expressing their meaning in words

When a calculation is specified, work at least two examples

Look for statements that imply certainty, then ask for proof keys; always, every, all, none, never

Search behind certainty statements—be sure restrictions or limitations are realistic