



Kuliah

***Rekayasa Perangkat Lunak  
(Software Engineering)***

Bagian 2

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

**Chapter 8**

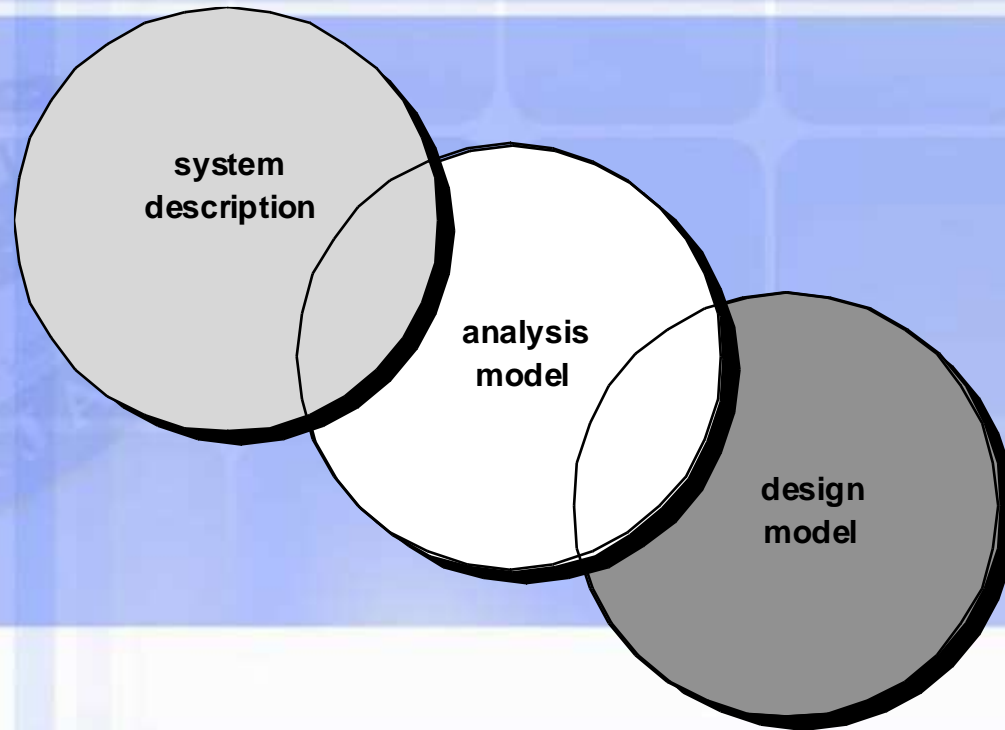
**Pemodelan Analisis**



# 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 konsideran 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* (lantai 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**