

REKAYASA PERANGKAT LUNAK

Danang Wahyu Utomo

danang.wu@dsn.dinus.ac.id

+6285 740 955 623

RENCANA KEGIATAN PERKULIAHAN SEMESTER

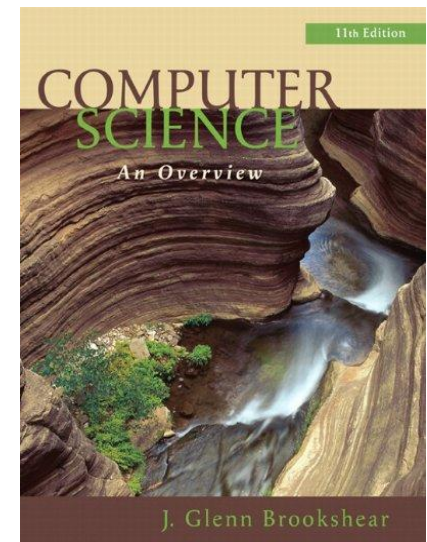
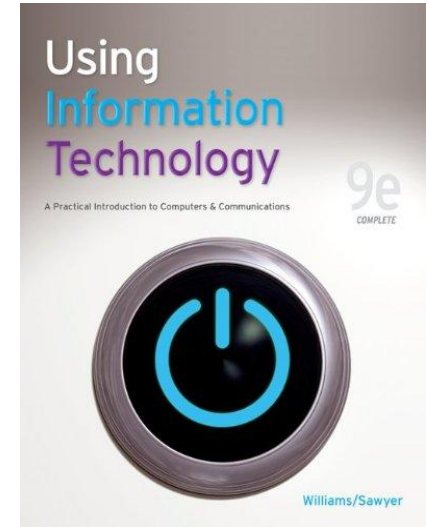
W	Pokok Bahasan
1	Pengenalan Teknologi Informasi
2	Konsep Sistem Komputer & Pengenalan Perangkat Keras
3	
4	Data Storage
5	Perangkat Lunak
6	
7	Data dan Informasi
8	Ujian Tengah Semester

W	Pokok Bahasan
9	Komputasi Pemrograman
10	
11	Rekayasa Perangkat Lunak
12	Komunikasi data & Jaringan Komputer
13	
14	Etika dan dampak sosial teknologi informasi
15	Teknologi Terkini / Advance Topik
16	Ujian Akhir Semester



Reference

- ▶ Bruce K William, Stacey C. Sawyer – Using Information Technology :A Practical Introduction to Computers & Communications 9th Edition (2010)
- ▶ J. Glenn Brookshear – Computer Science :An Overview 11th Edition (2011)



Content

The Software Engineering Discipline

The Software Life Cycle

Software Engineering Methodologies

Tools

Quality Assurance



The Software Engineering Discipline

- ▶ Software engineering adalah cabang computer science dengan prinsip pengembangan system software yang besar dan kompleks
- ▶ Pengembangan sistem membutuhkan upaya banyak orang
- ▶ Software engineering terdiri dari personel dan project management yang berkaitan dengan business management
- ▶ Software engineering membantu dalam pemilihan device yang kompleks

The Software Engineering Discipline

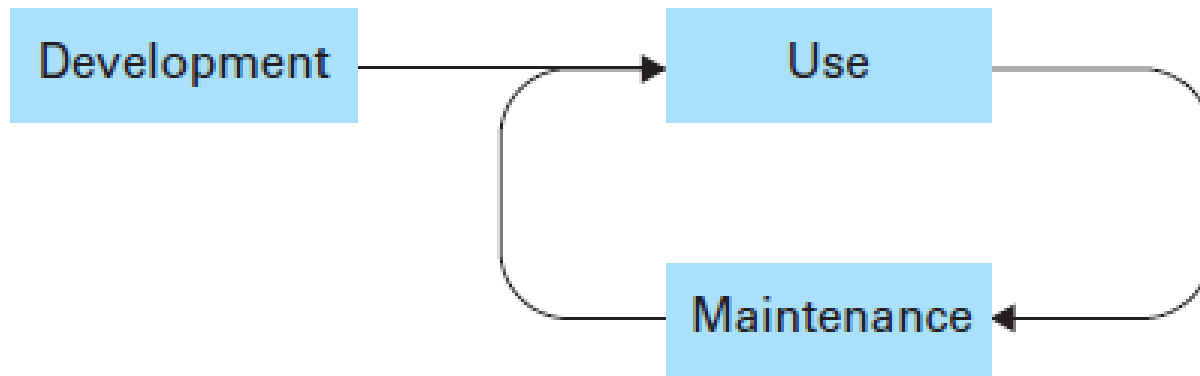
- ▶ Problem selama pengembangan sistem :
 - How can you **estimate** the cost in time, money and other resource to complete the project ?
 - How can you divide the project into manageable pieces ?
 - How can you **ensure** that the pieces produced are **compatible** ?
 - How can those working on the various pieces **communicate**?
 - How can you **measure progress**?

The Software Engineering Discipline

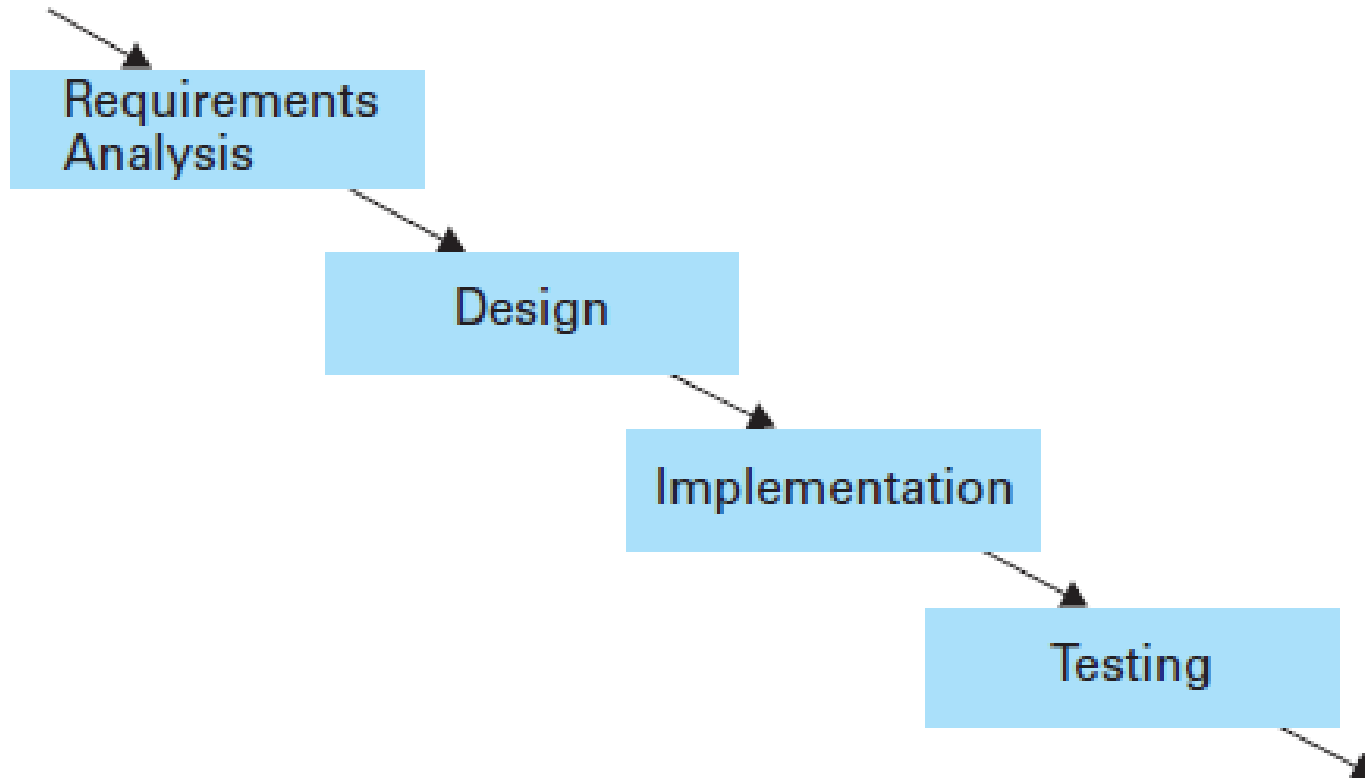
- ▶ Penelitian dalam software engineering saat ini terdiri dari dua level : **practitioners** dan **theoreticians**
- ▶ Practitioners bekerja ke arah pengembangan teknik untuk aplikasi
- ▶ Theoreticians bekerja berdasarkan prinsip dan teori dimana teknik lebih stabil dapat dibangun nantinya

The Software Life Cycle

- ▶ Konsep dalam software engineering adalah software life cycle
- ▶ Software life cycle menggambarkan bahwa dalam pengembangan software masuk ke dalam sebuah cycle yang digunakan (**development, use, maintenance**)



The Traditional Development Phase



The Traditional Development Phase

- ▶ Tahap Requirement Analysis :
 - Menganalisa kebutuhan *user*
 - Negoisasi dengan stakeholder mengenai kebutuhan, harga dan kemungkinan yang terjadi pada sistem
 - Membangun seperangkat kebutuhan untuk identifikasi apa saja fitur yang ada dalam sistem software

The Traditional Development Phase

- ▶ Tujuan Requirement Analysis :
 - Menentukan apa saja kebutuhan sistem yang diusulkan
 - Identifikasi kebutuhan sistem yang diusulkan
 - Mendefinisikan bagaimana user berinteraksi dengan sistem

The Traditional Development Phase

▶ Tahap Design :

- Membuat rancangan untuk system yang diusulkan
- Fokus pada pengembangan solusi
- Fokus pada “Bagaimana sistem akan melakukannya” ?
- Hasil dari design adalah tahapan struktur sistem secara detail yang akan dikonversikan ke program
- Diagram dan pemodelan penting dalam tahapan desain

The Traditional Development Phase

▶ Tahap Implementasi :

- Implementasi melibatkan penulisan program, pembuatan file, dan database
- Programmer memiliki beban pada penulisan program dimana sebelumnya telah didesain oleh *software analyst*
- Programmer menerjemahkan hasil dari **design** menjadi suatu program (aplikasi)

The Traditional Development Phase

▶ Tahap Testing :

- Proses debug program dan konfirmasi apakah software sesuai dengan spesifikasi kebutuhan software
- Testing digunakan untuk mengetahui kualitas dari software
- Beberapa kesalahan mungkin tidak terdeteksi pada saat tahap implementasi





How the customer explained it



How the Project Leader understood it



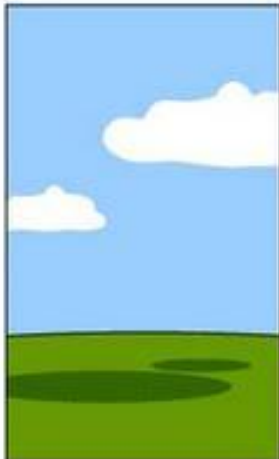
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



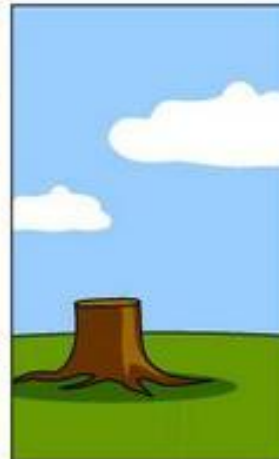
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed



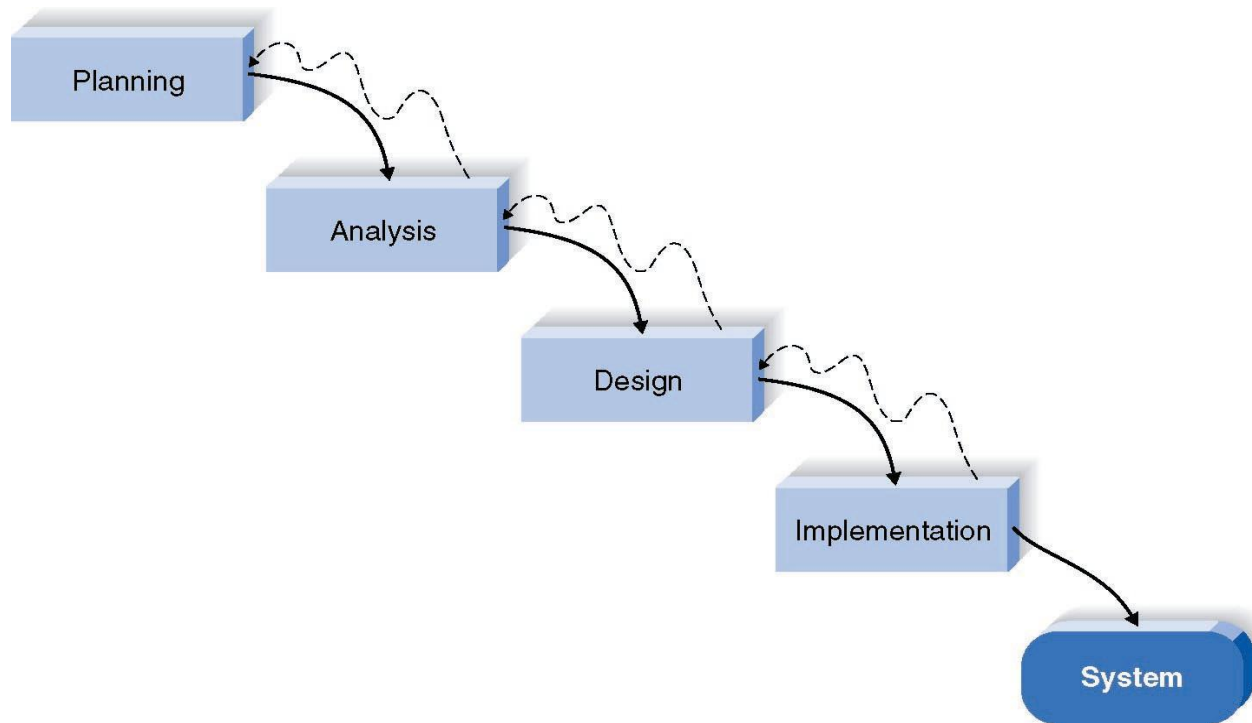
Software Engineering Methodologies

- ▶ Waterfall Model
- ▶ Prototyping
- ▶ RAD
- ▶ Extreme Programming (XP)



Waterfall

- ▶ Mengidentifikasi kebutuhan jauh sebelum pemrograman dimulai
- ▶ Meminimalkan perubahan kebutuhan selama project



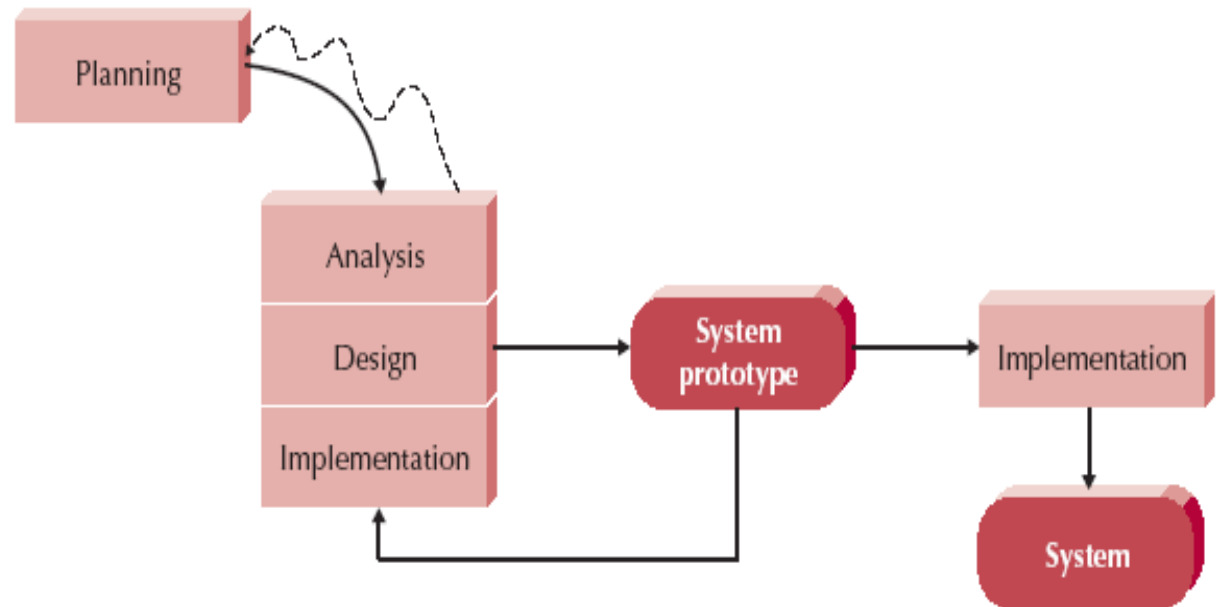
Waterfall

▶ Tahap Planning :

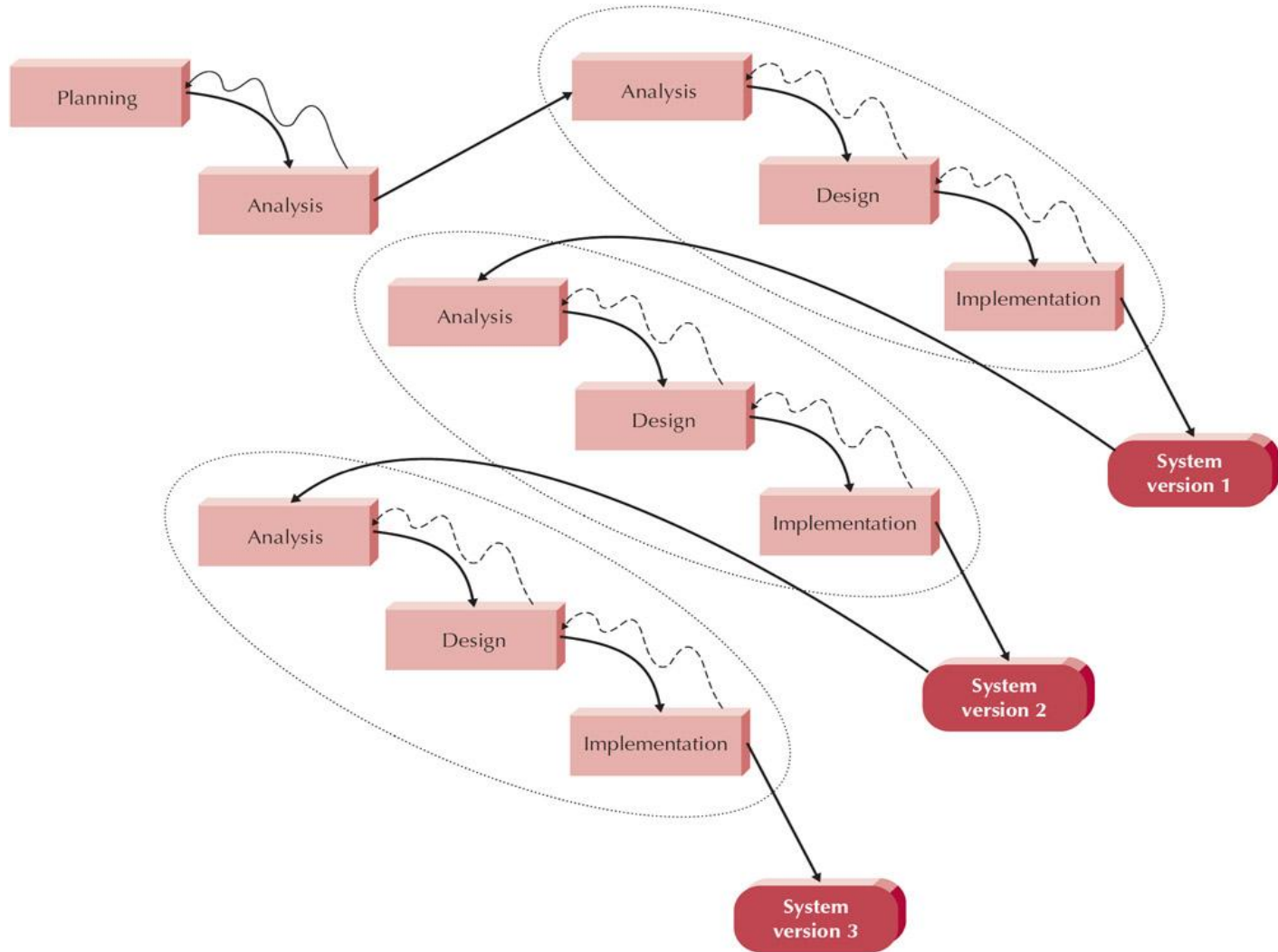
- Dalam form permintaan sistem, menjelaskan ringkasan dari kebutuhan bisnis
- Analisa kemungkinan aspek dari project yang diusulkan
- Bagaimana tim project berjalan selama pengembangan sistem
- Tahap Analysis
- Tahap Design
- Tahap Implementation

Prototyping

- ▶ Sangat cepat untuk mendefinisikan kebutuhan
- ▶ Sangat cepat menyediakan sistem
- ▶ User dapat langsung berinteraksi dengan prototype untuk mengetahui apa yang dapat dan tidak dilakukan oleh sistem



RAD



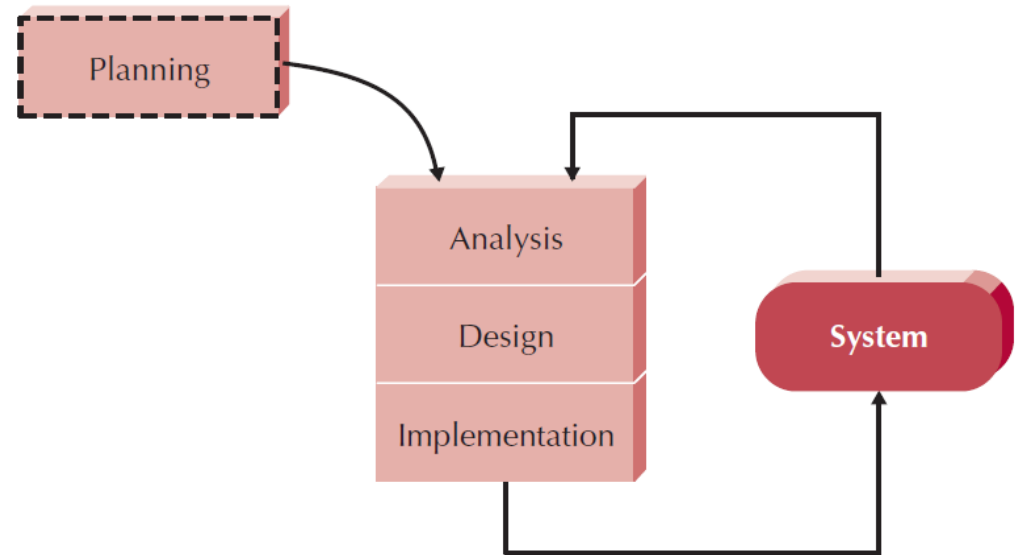
RAD

- ▶ Pemahaman yang lebih baik terhadap sistem dan revisi langsung dengan membawa sistem lebih cepat apa yang diinginkan user
- ▶ Problem dari RAD adalah mengelola keinginan pengguna terhadap kemungkinan perubahan yang terjadi



Extreme Programming (XP)

- ▶ XP terdiri dari empat nilai inti : komunikasi, kesederhanaan, umpan balik, dan keberanian
- ▶ Pengembang harus menyediakan umpan balik secara cepat
- ▶ Pengembang harus membuat perubahan tambahan untuk perkembangan sistem



Criteria for Selecting a Methodology

Ability to Develop Systems	Structured Methodologies			RAD Methodologies		Agile Methodologies
	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP
With Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent
With Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Poor
That Are Complex	Good	Good	Good	Poor	Excellent	Poor
That Are Reliable	Good	Good	Good	Poor	Excellent	Good
With a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent
With Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Good

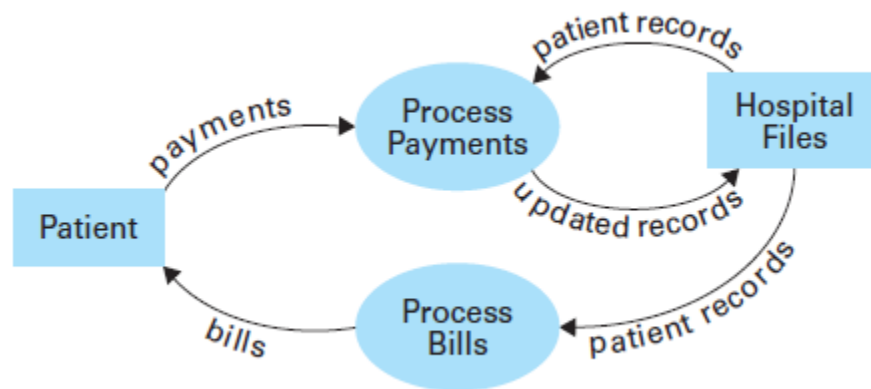


Tools

- ▶ Tool digunakan untuk membuat pemodelan selama tahapan analisis dan desain
- ▶ Pemodelan bertujuan untuk menggambarkan bagaimana data berpindah melalui sistem, meningkatkan komunikasi antara stakeholder dan software engineer
- ▶ Contoh : Dataflow Diagram, UML

Dataflow Diagram

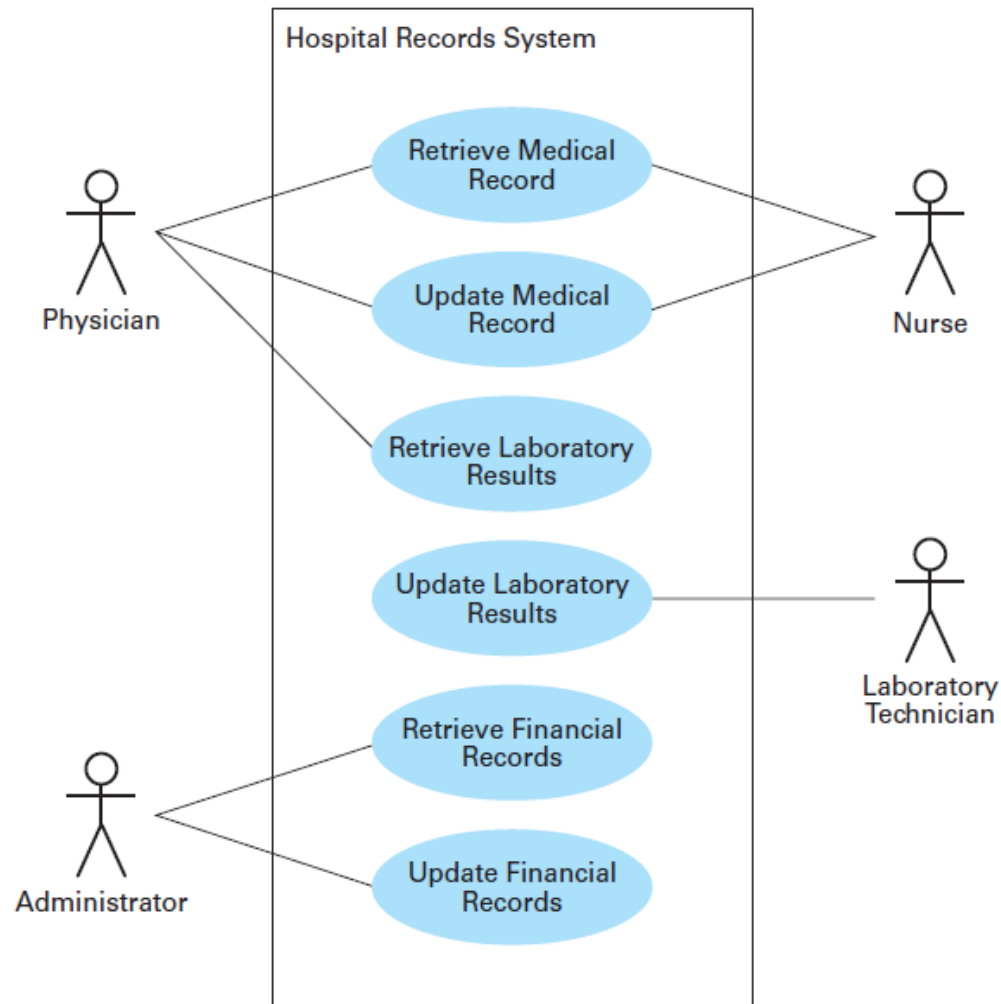
- ▶ Dataflow diagram menggambarkan informasi yang diperoleh dari dataflow
- ▶ Memudahkan untuk memahami sistem yang diusulkan selama tahap analisis
- ▶ Identifikasi prosedur selama tahap desain



UML (Unified Modeling Language)

- ▶ Sebuah tool yang dikembangkan dengan paradigma Object – Oriented
- ▶ Contoh : Use Case Diagram, Class Diagram
- ▶ Sama seperti DFD, use case menggambarkan interaksi antara user dan sistem
- ▶ Class diagram menggambarkan struktur kelas dan hubungan antar kelas

Use Case Diagram



Class Diagram

