



# Model Perancangan Perangkat Lunak



# Tujuan

- ▶ Menjelaskan definisi pengembangan sistem dan fase dan kegiatan pada system development lifecycle (SDLC)
- ▶ Menjelaskan perbedaan antara model, teknik, dan metodologi pengembangan sistem
- ▶ Menggambarkan 2 pendekatan yang digunakan untuk mengembangkan sistem informasi : metode terstruktur dan metode object-oriented
- ▶ Menggambarkan beberapa variasi dari system development life cycle (SDLC)
- ▶ Menggambarkan beberapa fitur kunci yang sedang trend saat ini dalam pengembangan sistem : spiral model, extreme programming (XP), dan Rational unified process (RUP)
- ▶ Menjelaskan bagaimana computer-aided system engineering (CASE) tools digunakan dalam pengembangan sistem



# Pengembangan Sistem

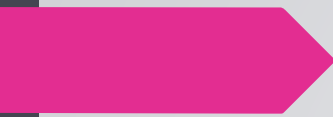


- ▶ Pengembangan sistem informasi sering disebut sebagai proses pengembangan sistem (*System Development*)
- ▶ Pengembangan sistem didefinisikan sebagai menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang ada
- ▶ Alasan Pengembangan Sistem :
  1. Adanya permasalahan
    - a. Adanya ketidakberesan
    - b. Pertumbuhan Organisasi
  2. Untuk meraih kesempatan-kesempatan
  3. Adanya instruksi (pimpinan, pemerintah)



# Pengembangan Sistem

- ▶ Sebenarnya untuk menghasilkan sistem informasi tersebut terdiri dari:
  - *System Analysis*: upaya mendapatkan gambaran bagaimana sistem bekerja dan masalah-masalah apa saja yang ada pada sistem
  - *System Development*: langkah-langkah mengembangkan sistem informasi yang baru berdasarkan gambaran cara kerja sistem & permasalahan yang ada



# Metode Pengembangan Sistem

- ▶ Banyak metode pengembangan sistem yang tersedia
- ▶ Metode yang paling dikenal disebut juga sebagai System Development Life Cycle (SDLC) atau sering juga disebut sebagai *Water Fall Method*
- ▶ Alternatif metode lain Prototyping : CASE tools, Joint Application Design (JAD), Rapid Application Development (RAD), Agile Methodologies, eXtreme Programming

# System Development Lifecycle (SDLC)

- ▶ SDLC merupakan metode pengembangan sistem paling tua
- ▶ Sangat cocok untuk pengembangan sistem yang besar
- ▶ Tidak sesuai atau tidak terlalu disarankan untuk *small scale project* karena:
  - Banyak memerlukan sumber daya
  - Tidak fleksibel
  - Sulit untuk melakukan perubahan aplikasi dengan pengambilan keputusan yang cepat
- ▶ SDLC lebih dari sekedar fase
  - ▶ Prinsip Manajemen
  - ▶ Perencanaan dan Pengawasan
  - ▶ Pengorganisasian dan Penjadwalan
  - ▶ Penyelesaian Masalah



# SDLC

## ► Keuntungan:

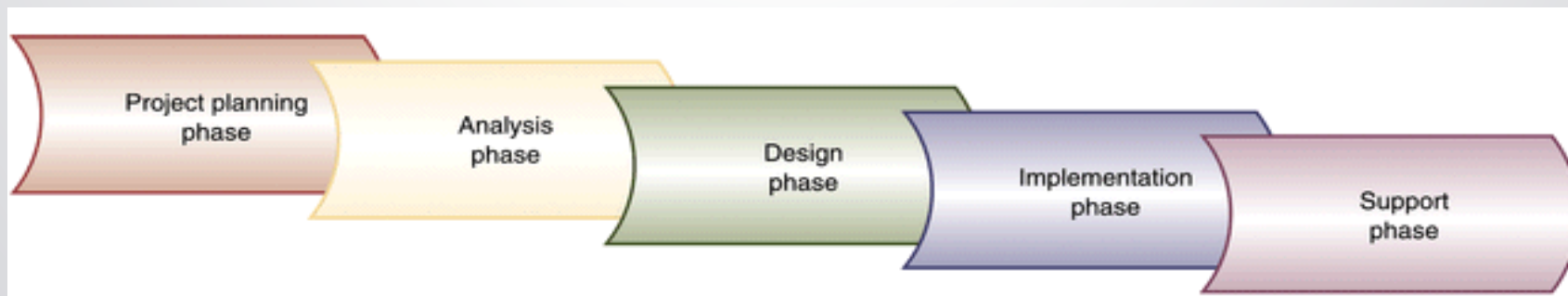
- Adanya laporan setiap akhir fase sehingga memudahkan adanya kontrol / pengawasan
- Mudah melakukan dokumentasi
- Dokumentasi secara formal sehingga memudahkan penelusuran kembali terhadap kebutuhan bisnis

## ► Kelemahan:

- Pengguna mendapatkan produk sesuai dengan pemahaman pengembang, yang belum tentu sesuai kebutuhan
- Dokumentasi mahal dan menghabiskan waktu dalam pembuatan, dan selalu berubah / mengalami perbaikan

# Systems Development Lifecycle (SDLC)

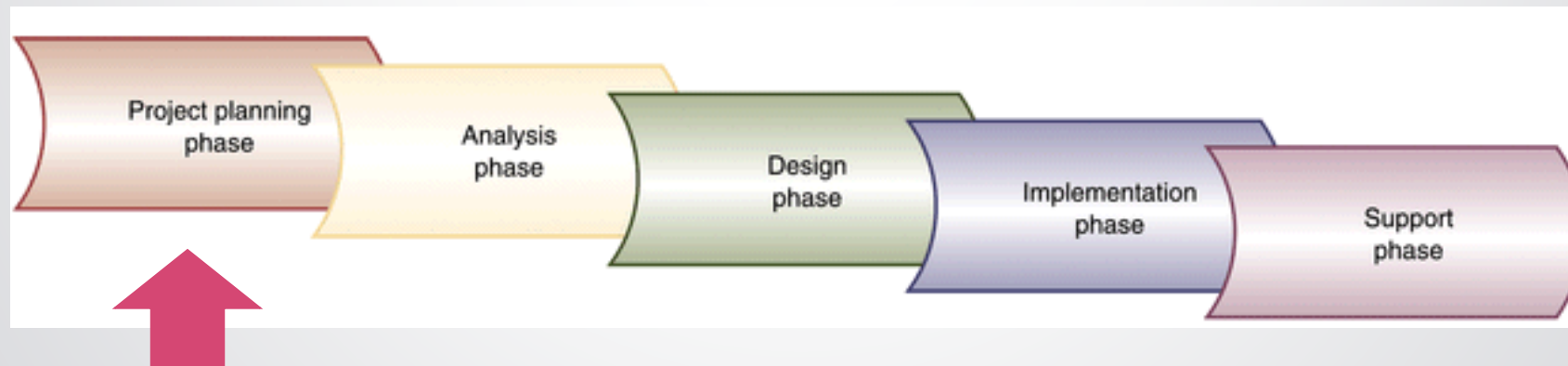
- ▶ SDLC terdiri dari 5 fase
  - ▶ Masing masing fase terdiri dari aktivitas yang saling terkait / berhubungan
- ▶ 3 Aktivitas utama
  - ▶ Analisa: memahami kebutuhan bisnis
  - ▶ Desain: membuat konsep solusi pengembangan sistem berbasis komputer
  - ▶ Implementasi: konstruksi / pembuatan, testing, dan instalasi
- ▶ 2 Fase tambahan
  - ▶ Perencanaan Proyek
  - ▶ Support / Dukungan





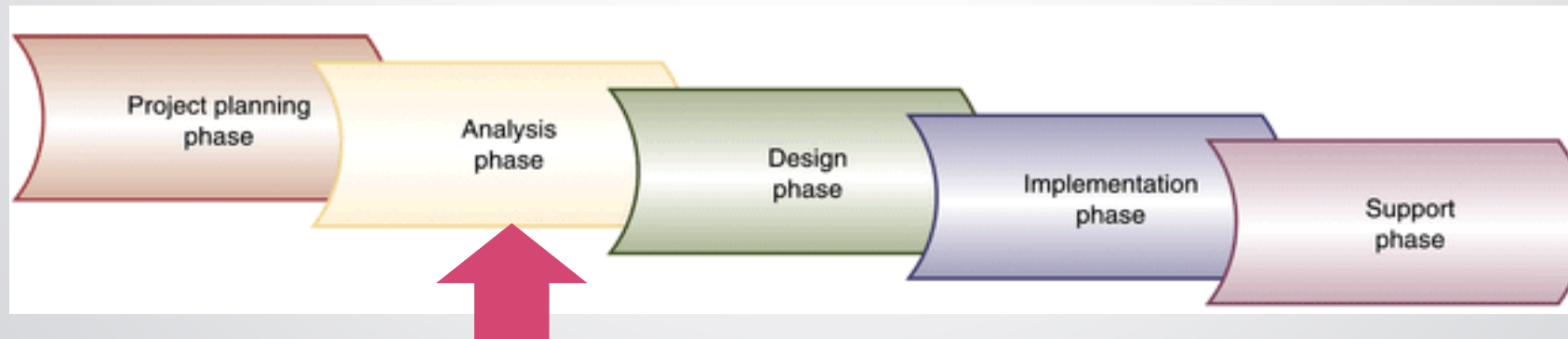
# Fase Perencanaan

- ▶ Mendefinisikan Masalah
- ▶ Mengkonfirmasi kelayakan proyek
- ▶ Membuat jadwal proyek
- ▶ Menentukan staff yang terlibat dalam proyek
- ▶ Memulai proses pengembangan proyek



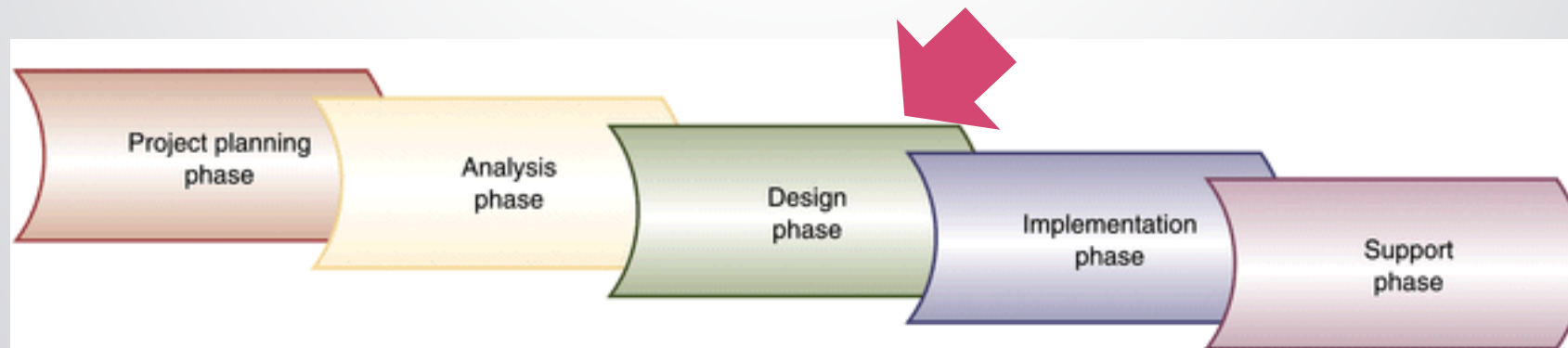
# Fase Analisa

- ▶ Mengumpulkan informasi
- ▶ Mendefinisikan kebutuhan - kebutuhan sistem
- ▶ Membangun prototipe yang sesuai atau memenuhi kebutuhan sistem
- ▶ Menentukan prioritas kebutuhan sistem
- ▶ Membuat prototipe atas prioritas dan melakukan evaluasi terhadap alternatif yang dipilih
- ▶ Mereview rekomendasi terhadap pihak manajemen



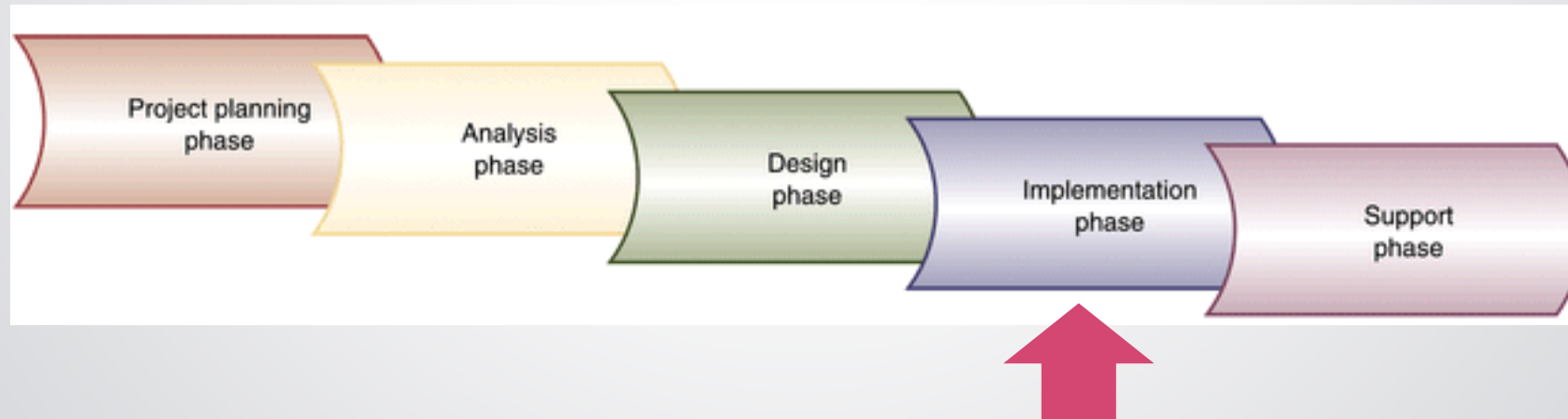
# Fase Desain

- ▶ Desain Level Tinggi (Arsitektur Sistem)
  - ▶ Desain dan integrasi jaringan
  - ▶ Desain arsitektur aplikasi
- ▶ Desain Level Rendah
  - ▶ Desain user interface
  - ▶ Desain sistem interface
  - ▶ Desain dan integrasi database
  - ▶ Prototype desain secara lengkap
  - ▶ Desain dan integrasi pengawasan sistem



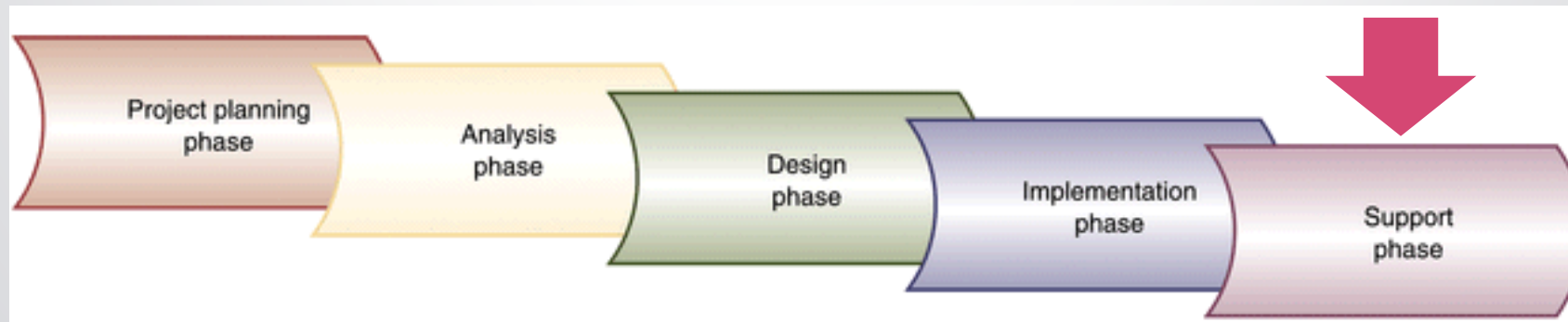
# Fase Implementasi

- ▶ Membangun komponen - komponen perangkat lunak
- ▶ Melakukan verifikasi dan pengujian
- ▶ Mengkonversi data
- ▶ Melakukan training user dan mendokumentasikan sistem
- ▶ Menginstall sistem



# Fase Support / Dukungan

- ▶ Memelihara Sistem
- ▶ Memperbaiki system
- ▶ Mendukung Pengguna
  - ▶ Help desk

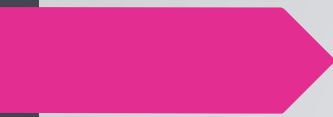


**Table 1-2** Products of SDLC Phases

<i>Phase</i>	<i>Products, Outputs, or Deliverables</i>
Planning	<p>Priorities for systems and projects; an architecture for data, networks, and selection hardware, and IS management are the result of associated systems;</p> <p>Detailed steps, or work plan, for project;</p> <p>Specification of system scope and planning and high-level system requirements or features;</p> <p>Assignment of team members and other resources;</p> <p>System justification or business case</p>
Analysis	<p>Description of current system and where problems or opportunities are with a general recommendation on how to fix, enhance, or replace current system;</p> <p>Explanation of alternative systems and justification for chosen alternative</p>
Design	<p>Functional, detailed specifications of all system elements (data, processes, inputs, and outputs);</p> <p>Technical, detailed specifications of all system elements (programs, files, network, system software, etc.);</p> <p>Acquisition plan for new technology</p>
Implementation	<p>Code, documentation, training procedures, and support capabilities</p>
Maintenance	<p>New versions or releases of software with associated updates to documentation, training, and support</p>

# Membangun vs. Membeli

- ▶ Aktifitas pada fase desain dan implementasi akan sangat berbeda antara mengembangkan sistem dengan cara membangun sistem dan membeli paket perangkat lunak yang sudah ada
- ▶ Membangun:
  - ▶ Pada Fase Desain Level rendah : mengembangkan spesifikasi User Interface, Database, Aturan Bisnis
  - ▶ Pada Fase Implementasi : Pengkodean dan Pengujian
- ▶ Membeli:
  - ▶ Pada Fase Desain Level rendah : Menilai dan memilih semua pilihan konfigurasi yang ada dan mengembangkan spesifikasi yang diperlukan yang akan ditambahkan dalam paket yang dibeli
  - ▶ Pada Fase Implementasi : Mengkonfigurasi dan melakukan pengujian terhadap konfigurasi yang telah dilakukan
- ▶ Untuk keduanya : Aktivitas Fase Desain Level Tinggi sama



# Masalah yang terjadi pada pendekatan Waterfall

- ▶ Persyaratan sistem "terkunci " setelah ditentukan (tidak dapat berubah)
- ▶ Keterlibatan pengguna terbatas (hanya dalam fase analisa kebutuhan sistem)
- ▶ Terlalu banyak fokus pada fase SDLC yang dapat merugikan praktek-praktek pengembangan sistem informasi



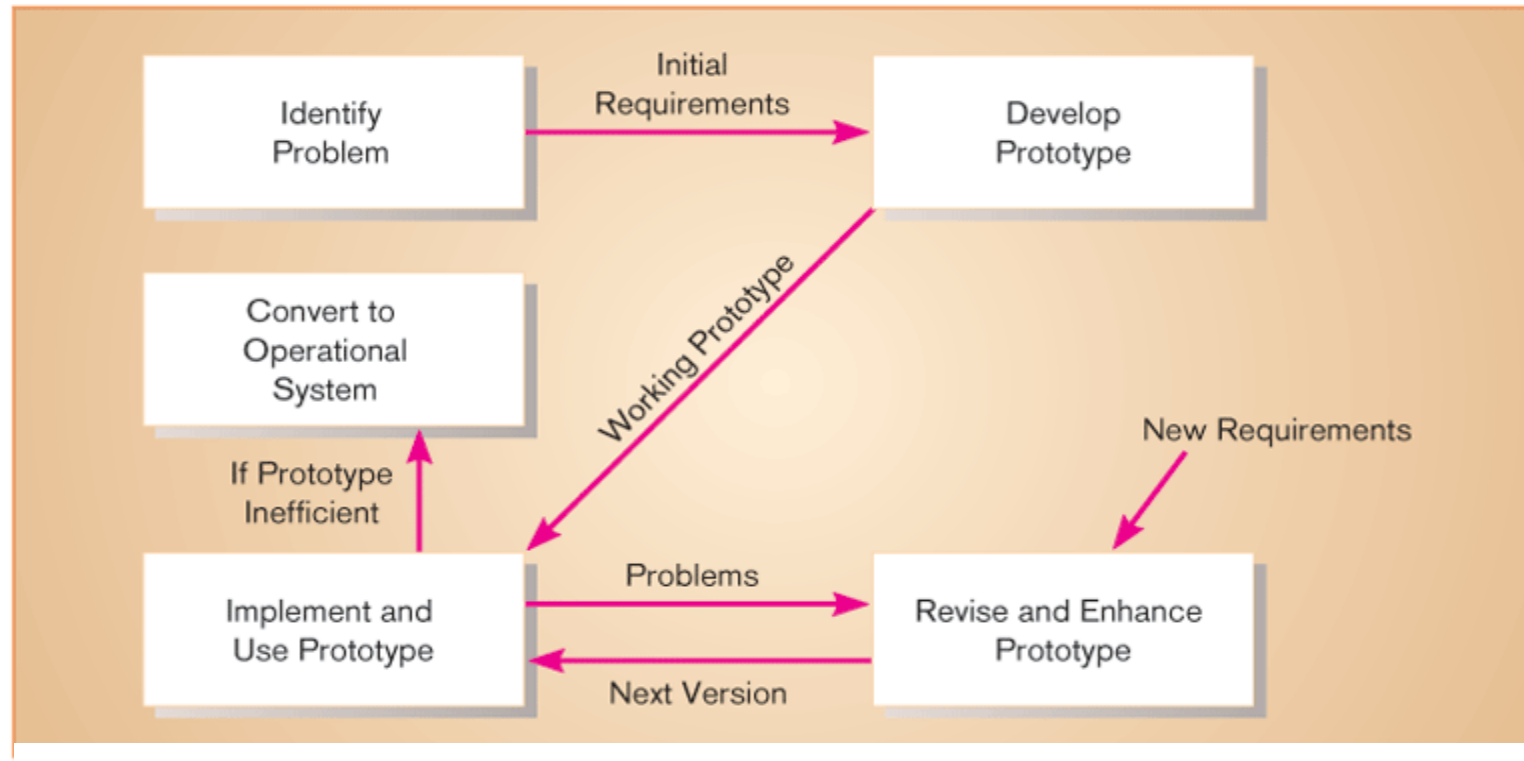


# Alternatif lain dari Tradisional Waterfall SDLC

- ▶ Prototyping
  - ▶ CASE tools
  - ▶ Joint Application Design (JAD)
  - ▶ Rapid Application Development (RAD)
  - ▶ Agile Methodologies
  - ▶ eXtreme Programming
- 

# Prototyping

**Figure 1-11** The prototyping methodology



## **Iteratif proses pengembangan:**

Persyaratan cepat dikonversi ke sistem kerja

Sistem terus direvisi

Kolaborasi tertutup antara pengguna dan analis




# CASE Tools



- ▶ Computer-Aided Software Engineering
- ▶ Software memberikan dukungan alat otomatis untuk pengembangan sistem
- ▶ Proyek kamus / buku kerja: sistem deskripsi dan spesifikasi diagram alat
- ▶ Contoh produk: Oracle Designer, Rational Rose



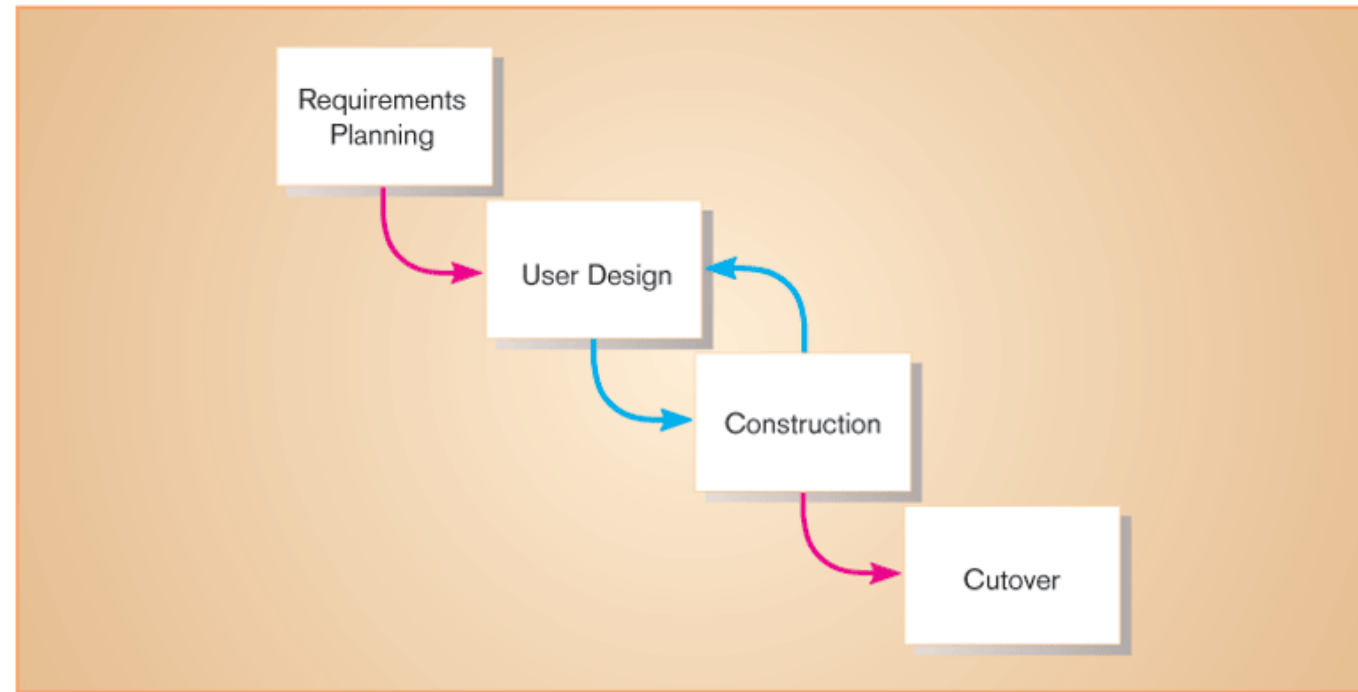
# Joint Application Design (JAD)

- ▶ Proses terstruktur yang melibatkan pengguna, analis, dan manajer
  - ▶ Membutuhkan beberapa hari sesi workgroup intensif
  - ▶ Tujuan: untuk menentukan atau meninjau persyaratan sistem
- 

# Rapid Application Development (RAD)

- ▶ Metodologi untuk mengurangi waktu desain dan implementasi
- ▶ Mencakup prototyping, JAD, CASE tools, dan generator kode

**Figure 1-12** RAD life cycle





# Agile Methodologies



- ▶ Termotivasi oleh pengakuan pengembangan perangkat lunak sebagai sesuatu yang tak terduga, dan dinamis
- ▶ Tiga prinsip kunci
  - ▶ Adaptif ketimbang prediksi
  - ▶ Tekankan orang ketimbang peran
  - ▶ Self-adaptif proses

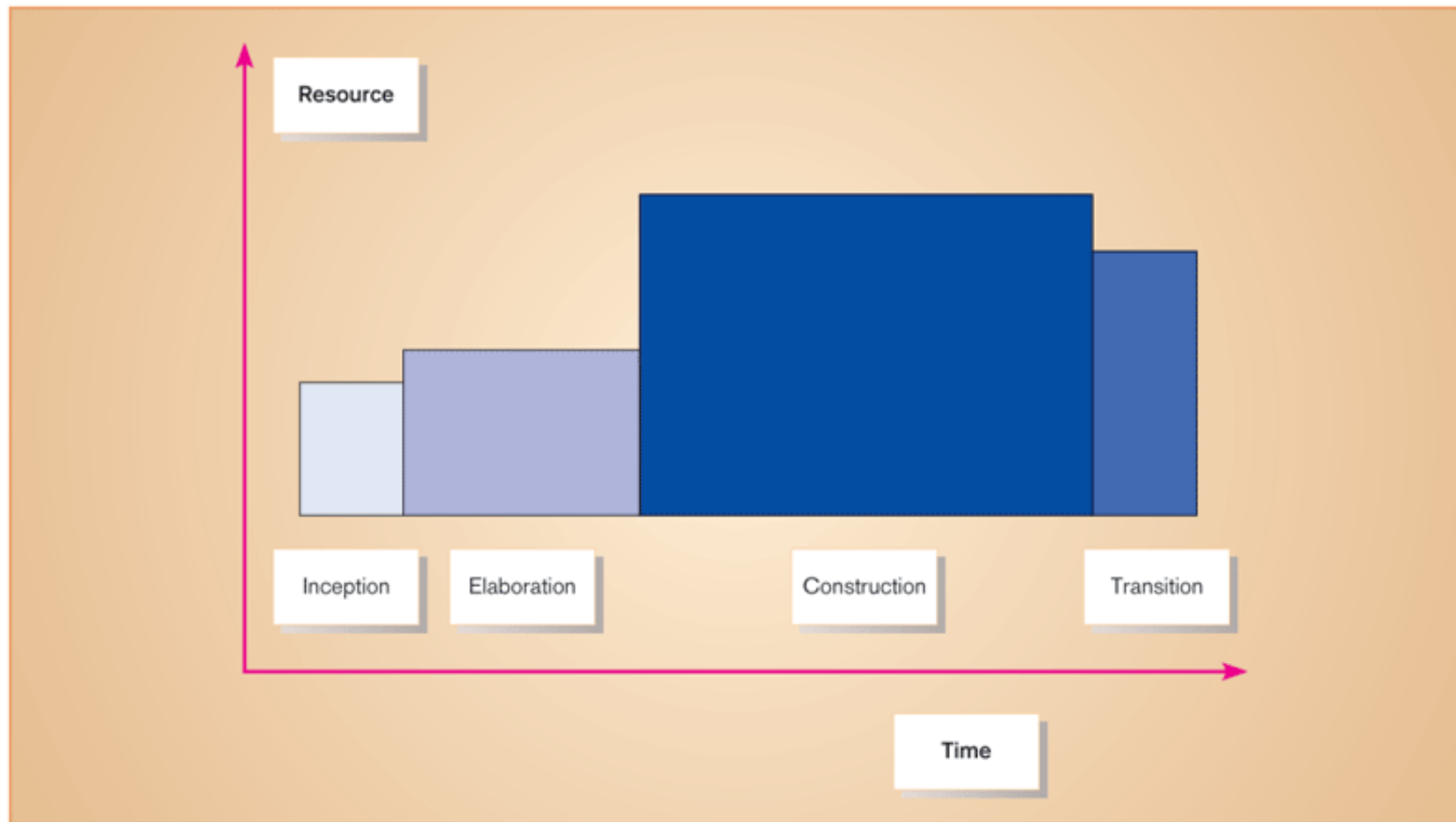


# eXtreme Programming

- Siklus pengembangan incremental, pendek
- Otomatis tes
- Dua orang tim pemrograman
- Pengkodean dan pengujian beroperasi bersamaan
- Keuntungan:
  - Komunikasi antara pengembang
  - Tinggi tingkat produktivitas
  - Tinggi kualitas kode

Rational Unified Process (RUP) melibatkan iteratif, pendekatan secara inkremental ke pengembangan sistem

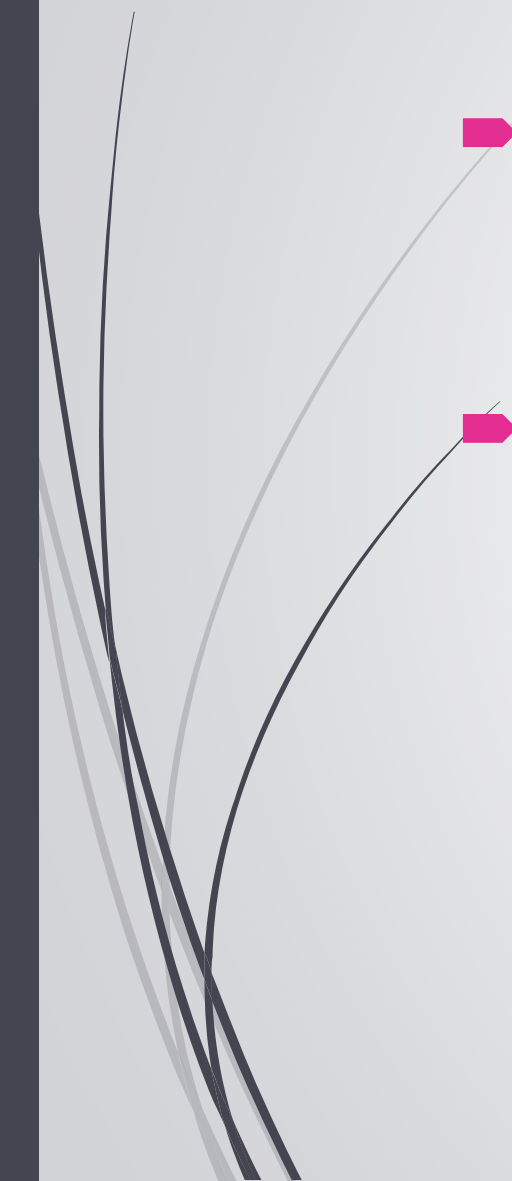
**Figure 1-13** Phases of OOSAD-based development







## Yang perlu dilakukan Analisis Sistem

- Analisis sistem perlu memilih pendekatan khusus untuk mengembangkan suatu sistem informasi - ada banyak alternatif.
  - Analisis perlu untuk memahami metodologi istilah, model, alat dan teknik
- 



# Metodologi

- ▶ Pedoman yang komprehensif untuk mengikuti setiap kegiatan untuk menyelesaikan SDLC
  - ▶ Koleksi model, peralatan, dan teknik
  - ▶ Dapat dikembangkan di rumah atau dibeli dari vendor eksternal
  - ▶ Varians yang luas di tingkat detail
- **Contoh:**
    - **Coopers and Lybrand Summit D**
    - **Accenture Method-1**
    - **Rational Unified Process (RUP)**



# Model

- Representasi aspek penting dari dunia nyata - abstraksi atau penyederhanaan
- Koleksi model yang dibutuhkan untuk berkomunikasi persyaratan sistem
- Banyak jenis model.  
Sistem Model - visual: gambar / diagram

## Some models of system components

Flowchart  
Data flow diagram (DFD)  
Entity-relationship diagram (ERD)  
Structure chart  
Use case diagram  
Class diagram  
Sequence diagram

## Some models used to manage the development process

PERT chart  
Gantt chart  
Organizational hierarchy chart  
Financial analysis models – NPV, ROI

# Tool

- ▶ Perangkat lunak pendukung yang membantu membuat model atau komponen proyek lainnya
- ▶ Dari program menggambar sederhana untuk CASE tools yang kompleks

Project management application  
Drawing/graphics application  
Word processor/text editor  
Computer-aided system engineering (CASE) tools  
Integrated development environment (IDE)  
Database management application  
Reverse-engineering tool  
Code generator tool

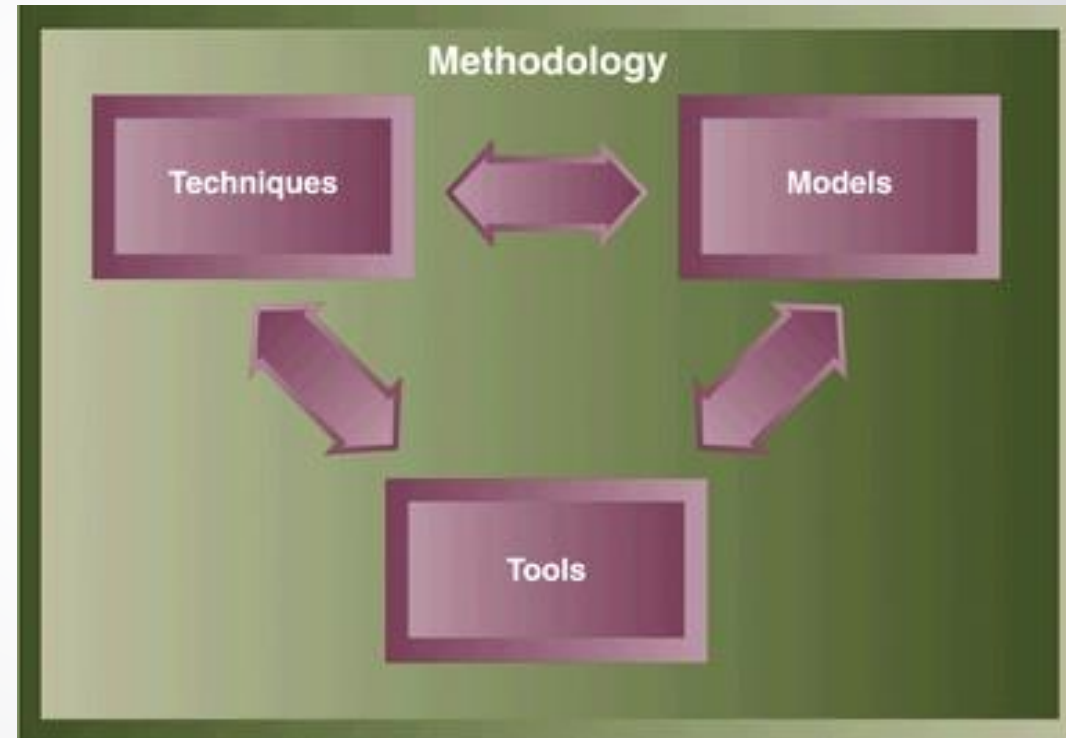
# Teknik

- ▶ Koleksi pedoman yang membantu analisis sistem kegiatan pembangunan lengkap atau tugas
- ▶ Berkisar dari saran umum ke Langkah-demi-langkah

Strategic planning techniques  
Project management techniques  
User interviewing techniques  
Data-modeling techniques  
Relational database design techniques  
Structured analysis technique  
Structured design technique  
Structured programming technique  
Software-testing techniques  
Object-oriented analysis and design techniques

# Metodologi , Model, Teknik dan Tool

- Metodologi merupakan kumpulan teknik yang digunakan untuk menyelesaikan semua kegiatan SDLC
- Teknik menghasilkan model (antara lain)
- Analisis menggunakan alat untuk menyelesaikan kegiatan dan untuk menghasilkan model





## Tugas 8

Sebutkan dan jelaskan serta gambarkan daur hidup pengembangan sistem informasi !