



ANALISIS DAN PERANCANGAN SISTEM INFORMASI (ANSI)

- ❑ Fitri Ayuning Tyas, S. Kom
- ❑ yas.0373@gmail.com
- ❑ 081804767700
- ❑ tyas-tamimy.com

PERENCANAAN PENGEMBANGAN SISTEM INFORMASI



Materi:

- A. Pengembangan Sistem Informasi
- B. Siklus Hidup Sistem
- C. System Life Cycle (SLC) & System Development Life Cycle (SDLC)
- D. Metode Pengembangan Sistem
 - 1. Waterfall Model
 - 2. Agile Model

A. PENGEMBANGAN SISTEM INFORMASI



- **Pengertian Pengembangan Sistem**

Pengembangan sistem (*system development*) dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada.

- Dilakukan dengan menggunakan metodologi (suatu proses standar yang diikuti oleh organisasi untuk melaksanakan seluruh langkah yang diperlukan untuk menganalisa, merancang, mengimplementasikan, dan memelihara sistem informasi)
- Metodologi klasik yang digunakan dikenal dengan istilah **SDLC** (*System Development Life Cycle*)

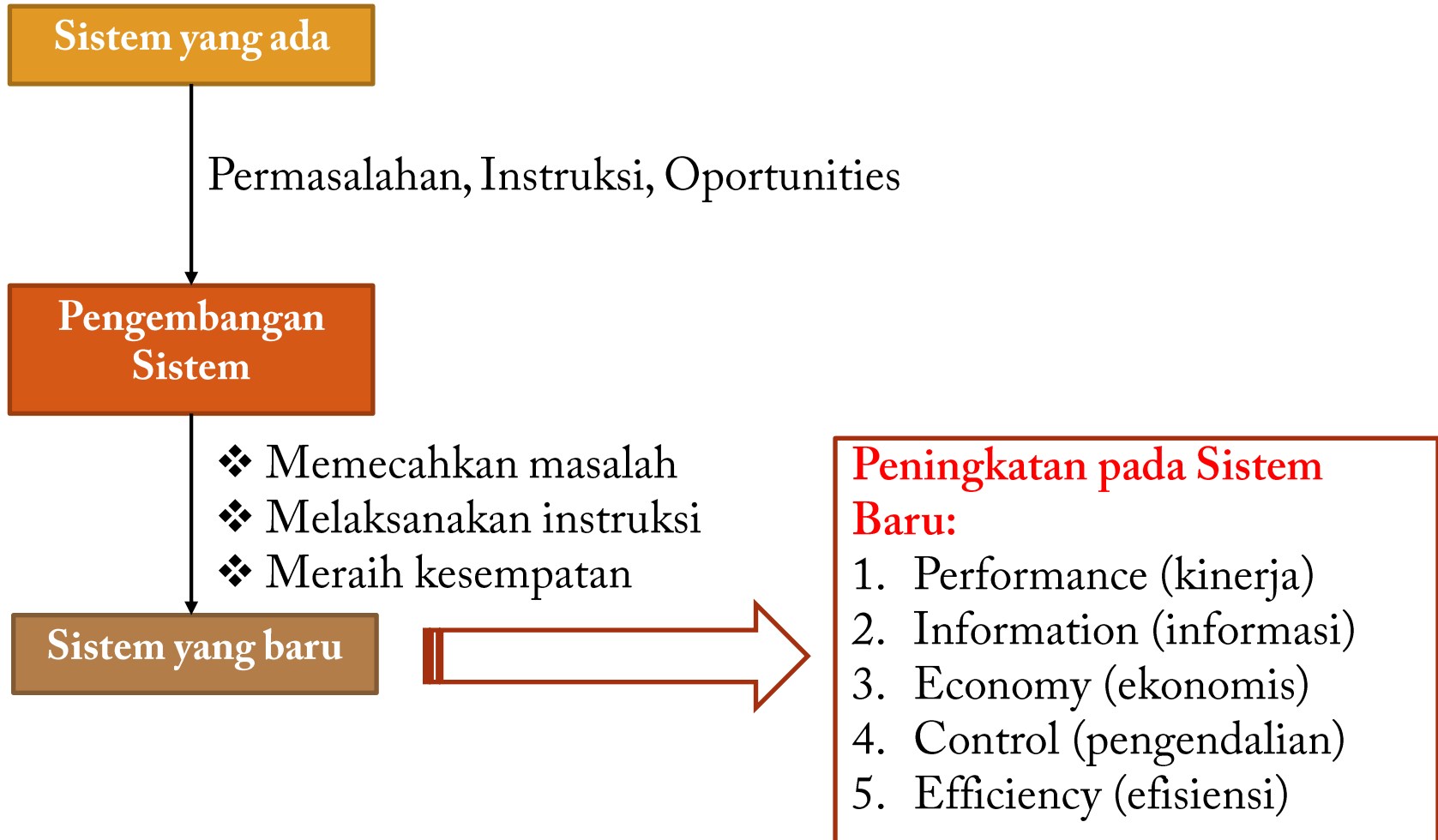
1. Alasan Pengembangan Sistem



Adanya permasalahan-permasalahan (problems) yang timbul di sistem yang lama yang dapat berupa :

- a) Ketidakberesan
- b) Pertumbuhan Organisasi
- c) Untuk meraih kesempatan-kesempatan (Opportunities)
- d) Adanya instruksi-instruksi (derivatives)

2. Proses Pengembangan Sistem



3. Prinsip Pengembangan Sistem



- Sistem yang dikembangkan adalah untuk manajemen
- Sistem yang dikembangkan adalah investasi modal yang besar
- Semua alternative yang ada harus diinvestigasi
- Investasi yang terbaik harus bernilai
- Sistem yang dikembangkan memerlukan orang yang terdidik
- Tahapan kerja dan tugas-tugas yang harus dilakukan dalam proses pengembangan sistem
- Proses pengembangan sistem tidak harus urut
- Jangan takut membatalkan proyek
- Dokumentasi harus ada untuk pedoman dalam pengembangan sistem

B. SIKLUS HIDUP SISTEM



- **Software Process** merupakan serangkaian kegiatan yang mengarah ke produksi produk perangkat lunak (Ian Sommerville, 2007)
- Terdapat 2 type software process:
 1. **Plan-driven processes** merupakan proses di mana semua kegiatan proses yang direncanakan terlebih dahulu dan kemajuan diukur terhadap rencana ini. Ex. Water Fall Model
 2. **In agile processes** merupakan perencanaan tambahan dan lebih mudah untuk mengubah proses yang mencerminkan perubahan kebutuhan pelanggan. Ex. Agile
- Perangkat lunak memiliki siklus hidup yang dikenal dengan siklus hidup perangkat lunak (**Software Life Cycle**)

1. Pentingnya SDLC (Dennis, 2005)

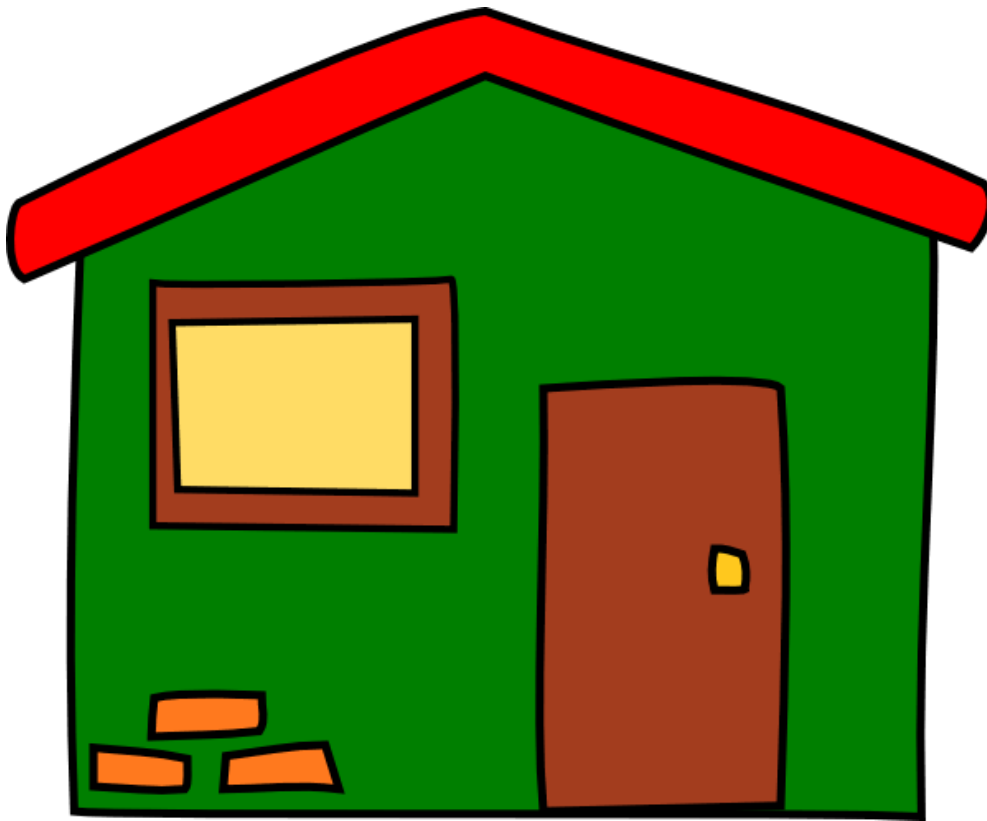


- Tahun 1996 suatu survey oleh *The Standish Group* menemukan bahwa **42%** dari semua *project* Sistem Informasi **diserahkan sebelum tuntas (*completion*)**
- Begitu pula studi dilakukan oleh *General Accounting Office* tahun 1996 menemukan hal yang sama sebesar **53%** *project Information System* di pemerintahan US
- Sebagian besar sistem diserahkan kepada pengguna dengan keterlambatan, biaya yang lebih dari perencanaan dan mempunyai fitur yang kurang dari yang sebenarnya

2. Ilustrasi



- *Bulding a system like building a house*



Membangun rumah membutuhkan perencanaan, analisa, design dan konstruksi/ pembangunan/ implementation

C. SYSTEM LIFE CYCLE (SLC)

&

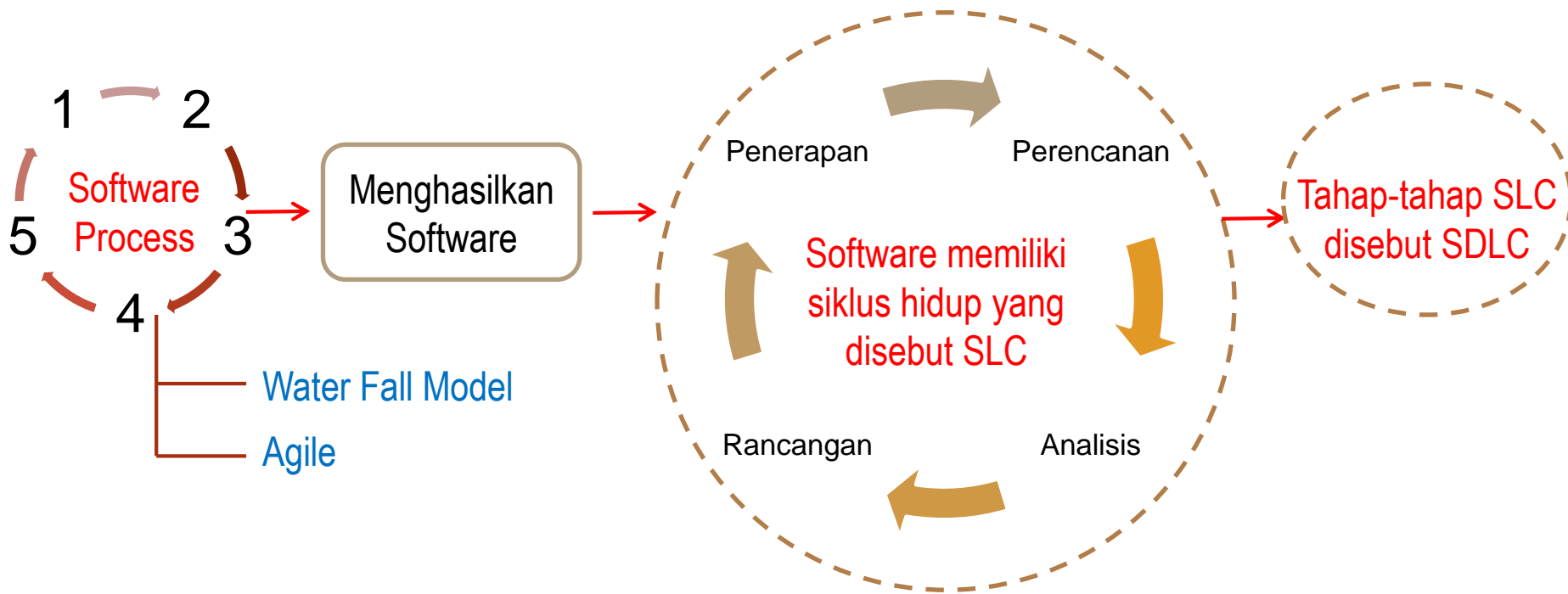
SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)

1. System Life Cycle (SLC) & System Development Life Cycle (SDLC)



- Siklus hidup perangkat lunak (**Software Life Cycle**) adalah urutan dari kegiatan yang ada di dalam sebuah pengembangan perangkat lunak (Gustafson, 2002)
- Software Development Process, juga dikenal sebagai **Software Development Life-Cycle** / siklus hidup pengembangan perangkat lunak, adalah struktur yang dikenakan pada pengembangan produk perangkat lunak dengan tahapan Perencanaan (Planning), Analisis (Analysis), Desain (Design), Penerapan (Implementation)

2. Software Proses, SLC & SDLC



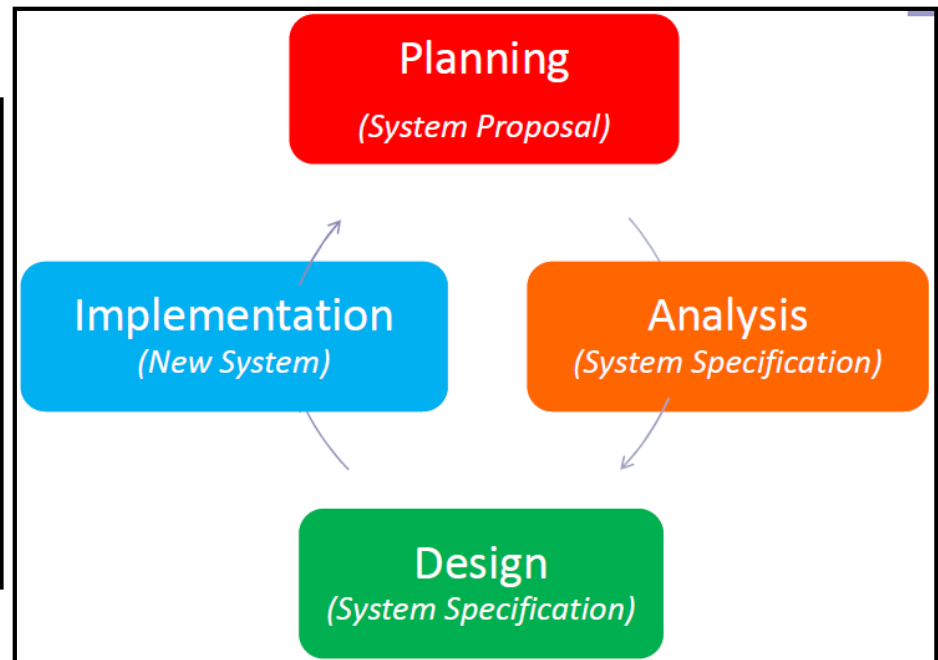
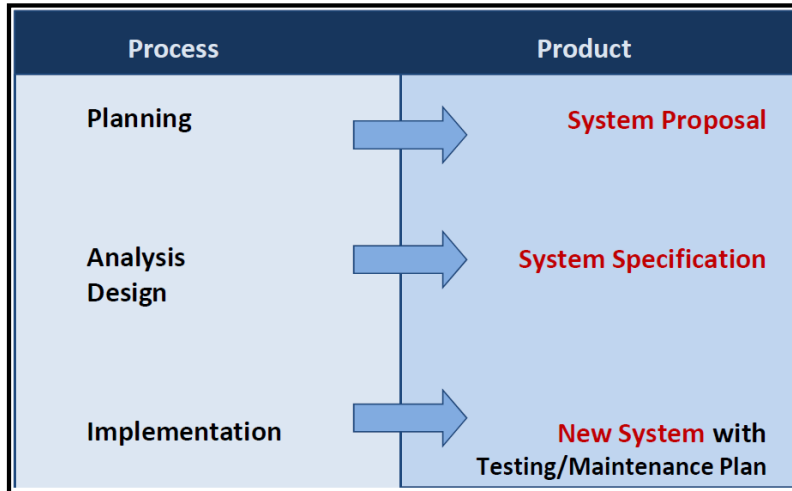
Software Life Cycle dan Software Process merupakan bagian dari siklus hidup pengembangan sistem (System Development Life Cycle)

3. Tahap-Tahap SDLC



- SDLC memiliki 4 tahapan mendasar (Dennis, 2005):
 1. Planning: **Why build the system?**
System request, feasibility analysis, project size estimation
 2. Analysis: **Who, what, when, where will the system be?**
Requirement gathering, business process modeling
 3. Design: **How will the system work?**
Program design, user interface design, data design
 4. Implementation: **System construction and delivery**
System construction, testing, documentation and installation

3. Tahap-Tahap SDLC (lanjut)



3.1. Tahap Perencanaan



1. Menyadari Masalah

Kepekaan pelaku sistem sangat membantu dalam pencetusan timbulnya masalah. Dengan kepekaan ini diharapkan dapat mendeteksi sedini mungkin terjadinya masalah dalam sistem.



Menyadari Masalah.

- keluhan pelanggan terhadap pelayanan .
- gaji sering terlambat dibayarkan.
- laporan sering terlambat.
- produktivitas kerja rendah.
- banyak pegawai yang menganggur.
- kegiatan yang tumpang tindih.
- biaya operasional yang tinggi.
- penyimpanan data yang tidak teratur.
- pekerjaan manajemen yang terlalu teknis.

3.2. Tahap Perencanaan



2. Mendefinikan Masalah

Menyadari adanya masalah, manajemen harus memahaminya sehingga dapat mengidentifikasi dan mengetahui letak permasalahannya.

3. Menentukan Tujuan Sistem

Dalam mengembangkan sistem, tujuan atau keinginan yang akan dicapai oleh sistem perlu untuk dirumuskan terlebih dahulu. Tujuannya agar dalam pengembangan sistem tidak bias tetapi terarah pada yang hendak dicapai.

4. Mengidentifikasi Kendala Sistem

Setiap pengoperasian sistem tidak bebas dari kendala baik pemakai, hasil maupun lingkungan luar sistem. Dengan mengidentifikasi kendala ini meminimalkan kesalahan dalam pembuatan sistem.

5. Membuat Studi Kelayakan

Studi kelayakan adalah tinjauan sekilas pada faktor-faktor utama yang akan mempengaruhi kemampuan sistem untuk mencapai tujuannya.

3.2. Tahap Perencanaan (lanjut)



Faktor-faktor yang perlu ditinjau:

- a. Teknis: Ketersediaan perangkat keras dan perangkat lunak dalam pemrosesan data.
- b. Pengembalian Ekonomis: Membandingkan keuntungan yang diperoleh dari penggunaan dengan biaya yang dikeluarkan untuk membangun sistem tersebut.
- c. Pengembalian Non Ekonomis: Menilai keuntungan pengembangan sistem yang diukur bukan dengan uang.
- d. Hukum dan Etika: Meninjau segi hukum dan etika, agar dapat dipastikan bahwa pengembangan sistem telah sesuai dengan norma hukum dan etika.
- e. Operasional: Meninjau sejauh mana para karyawan mendukung sistem yang baru.

3.4. Tahap Analisis Sistem



- Analisa Sistem bertujuan mengidentifikasi informasi yang dibutuhkan.
- Tahapan Analisis Sistem:
 1. Analisis Pendahuluan
 2. Penyusunan Usulan Pelaksanaan Analisis
 3. Pelaksanaan Analisis
 4. Penyusunan Laporan Hasil Analisis

3.5. Tahap Analisis Sistem (lanjut)



1. Analisis Pendahuluan

Bertujuan untuk mendapatkan gambaran tentang:

- luas pekerjaan yang akan dilakukan
 - hasil yang diinginkan
 - berapa lama pekerjaan ini dilakukan
- Pekerjaan yang dilakukan pada tahap ini adalah mengidentifikasi masalah agar lebih mengenal masalah dan mengumpulkan data data-data awal sehubungan dengan permasalahan.

3.5. Tahap Analisis Sistem (lanjut)



- **Alat yang digunakan:**
 - Struktur Organisasi & Job. Diskripsi
 - Prosedur kegiatan.
 - Dokumen yang digunakan
 - Laporan yang dihasilkan.
 - Aturan-aturan yang ada.
- **Identifikasi yang dilakukan:**
 - identifikasi penyebab masalah.
 - identifikasi titik keputusan.
 - identifikasi person kunci.

3.5. Tahap Analisis Sistem (lanjut)



2. **Penyusunan Usulan Pelaksanaan Analisis**

Bertujuan untuk mempertemukan pikiran antara pemakai sistem dengan analisis sistem mengenai rencana pengembangan sistem untuk memenuhi kebutuhan pemakai sistem.

- Dokumen Usulan Pelaksanaan analisis menjelaskan:
 1. alasan mengapa sistem dikembangkan
 2. persyaratan kinerja yang diharapkan dari sistem yang diusulkan
 3. batasan analisis sistem yang akan dilakukan
 4. informasi yang dibutuhkan
 5. sumber-sumber yang menyediakan informasi

3.5. Tahap Analisis Sistem (lanjut)



3. Pelaksanaan Analisis Sistem

Langkah langkah yang dilakukan analisis sistem:

- Menganalisis laporan/informasi yang dihasilkan sistem sekarang
- Menganalisis transaksi yang dilakukan:
 - unit organisasi yang terkait
 - form/dok yang digunakan
 - sistem otorisasi
- Mempelajari catatan pertama (jurnal), dengan tujuan untuk mempelajari apakah terdapat kelemahan dan kemungkinan pengembangan jurnal yang sekarang digunakan.

3.5 . Tahap Analisis Sistem (lanjut)



3. Pelaksanaan Analisis Sistem (lanjutan...)

➤ Menganalisis Hasil Penelitian.

Berdasarkan data-data yang diperoleh, tugas selanjutnya adalah menganalisis yaitu :

- Analisa Kelemahan sistem
- Analisa distribusi pekerjaan.
- Analisa pengukuran pekerjaan.
- Analisa Keandalan
- Analisa dokumen.
- Analisa laporan.
- Analisa teknologi.
- Analisa Kebutuhan Informasi.

3.5 . Tahap Analisis Sistem (lanjut)



3. Pelaksanaan Analisis Sistem (lanjutan...)

- Membuat rencana pengumpulan data:
 - jadwal
 - Penugasan tim
 - Persiapkan alat bantu (wawancara, daftar pertanyaan dan Observasi)
- Penyusunan Laporan Hasil Analisis, dokumen tertulis yang dibuat analis yang berisi temuan-temuan dari sistem yang berjalan saat ini

3.6. Tahap Desain Sistem



1. **Desain Sistem Secara Garis Besar**

Berdasarkan hasil analisis yang telah dilakukan, analis menawarkan alternatif desain sistem sesuai blok sistem berupa desain:

- keluaran
- masukan
- model
- basis data dan
- teknologi
- Pengendalian

2. **Evaluasi Sistem**

Analisis sistem mengkonsultasikan hasil rancangan kepada pemakai sistem

3.6. Tahap Desain Sistem (lanjut)



3. **Desain Final**

Hasil konsultasi/diskusi, dibuat laporan final desain sistem

4. **Desain Sistem Rinci**

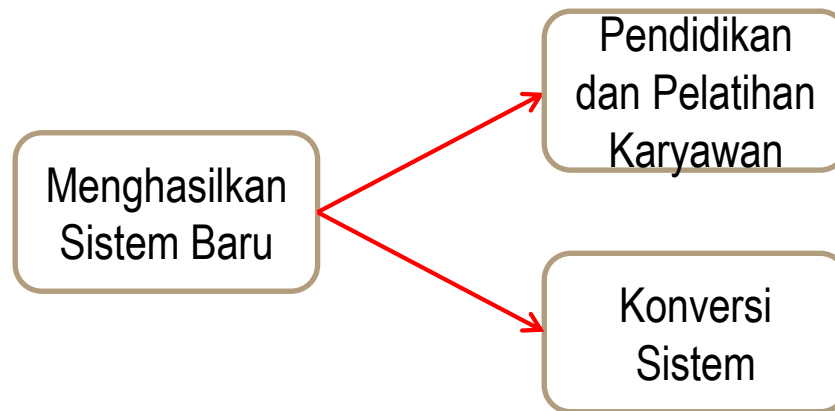
Saat pemikiran analis dan pemakai sistem bertemu dan tidak ada lagi masalah (sesuai hasil evaluasi sistem) maka analis menyusun desain secara rinci

5. **Laporan Rinci**

3.7. Tahap Implementasi

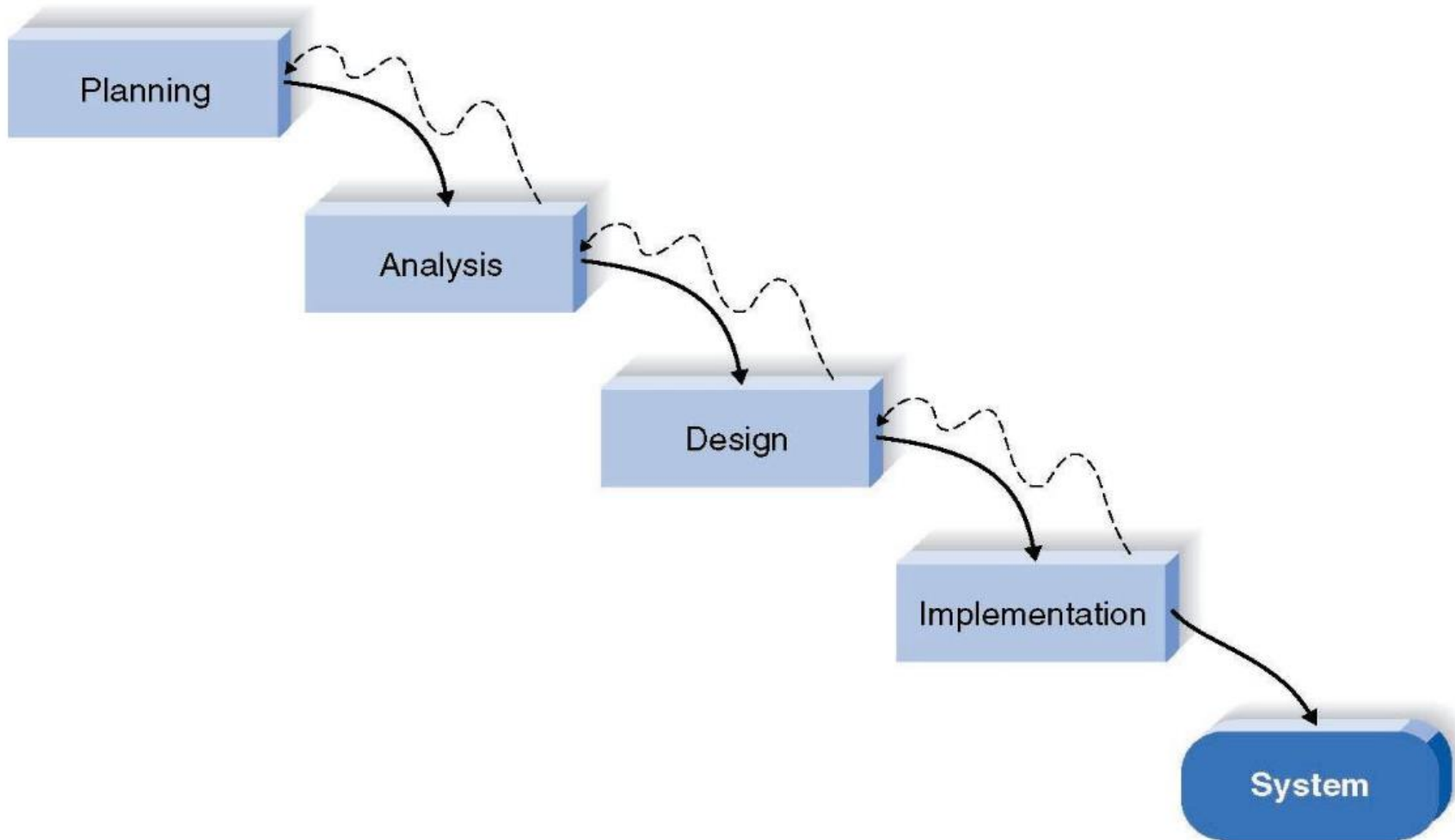


- **Implementasi sistem** adalah menjalankan sistem yang dikembangkan untuk mengganti sistem yang berjalan saat ini



D. METODE PENGEMBANGAN SISTEM

1. Waterfall Model



1. Waterfall (lanjut)



- “Classic Life Cycle” atau model Waterfall merupakan model yang paling banyak dipakai didalam Software Engineering (SE).
- Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, coding, testing / verification, dan maintenance.
- Disebut dengan waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

1.1. Kelebihan Waterfall Model



- Mudah untuk dipahami, mudah untuk digunakan
- Tahapan dipahami dengan baik
- Proses menjadi teratur
- Estimasi proses menjadi lebih baik
- Progress untuk setiap tahap dapat dilihat secara pasti
- Jadwal menjadi lebih menentu
- Kontrol management baik
- Bekerja baik ketika kualitas lebih penting dari biaya atau jadwal
- Pengidentifikasian system request yang lama sebelum memulai menuliskan kode (*programming*), meminimalisasi perubahan-perubahan yang terjadi

1.2. Kekurangan Waterfall Model



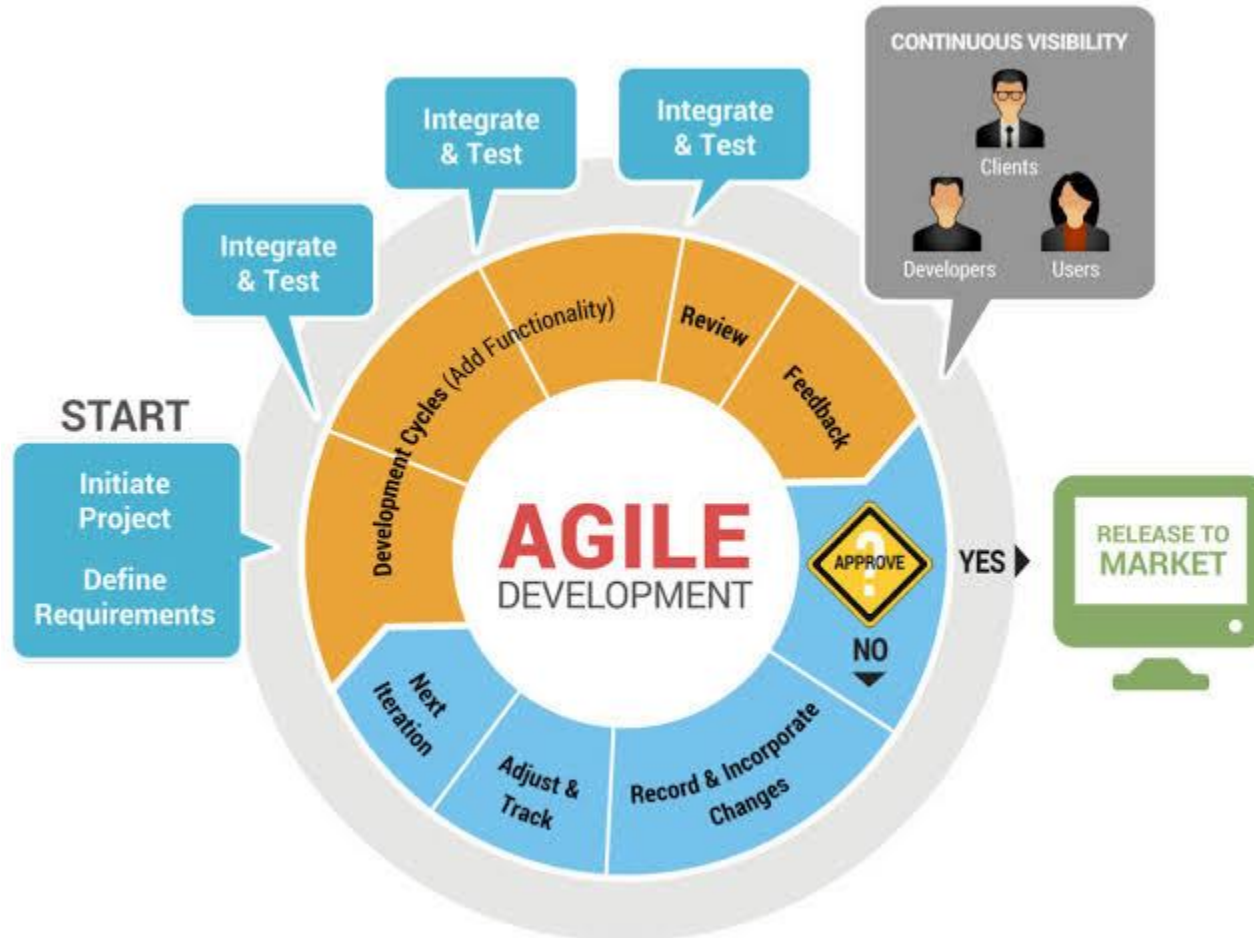
- Semua kebutuhan harus diketahui di awal, tapi jarang konsumen bisa memberikan kebutuhan secara lengkap di awal
- Sifatnya kaku, sehingga susah melakukan perubahan di tengah proses
- Waktu yang lama antara system proposal dan penyerahan sistem baru
- Design harus spesifik sebelum melakukan programming
- Kemungkinannya kecil bagi *customer* untuk dapat melihat preview sistem yang sedang dikerjakan

1.3. When to use the Waterfall Model?



- Kebutuhan telah diketahui dengan baik
- Definisi produk stabil
- Versi baru dari suatu sistem yang telah ada

2. Agile Model



2. Agile Model (lanjut)



- Agile merupakan adalah jenis pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dan pengembang terhadap perubahan dalam bentuk apapun.
- Dalam Agile Software Development interaksi dan personel lebih penting dari pada proses dan alat, software yang berfungsi lebih penting daripada dokumentasi yang lengkap, kolaborasi dengan klien lebih penting dari pada negosiasi kontrak, dan sikap tanggap terhadap perubahan lebih penting daripada mengikuti rencana.

2.1. Kelebihan Agile Model



- Meningkatkan kepuasan kepada klien
- Pembangunan system dibuat lebih cepat
- Mengurangi resiko kegagalan implementasi software dari segi non-teknis
- Jika pada saat pembangunan system terjadi kegagalan, kerugian dari segi materi relative kecil.

2.2. Kekurangan Agile Model



- Developer harus selalu siap dengan perubahan karena perubahan akan selalu diterima.
- Agile tidak akan berjalan dengan baik jika komitmen tim kurang.
- Tidak cocok dalam skala tim yang besar (>20 orang).
- Perkiraan waktu release dan harga perangkat lunak sulit ditentukan.