

# Metodologi Pengembangan Sistem

Analisis dan Perancangan Sistem Informasi  
[erik@uniku.ac.id](mailto:erik@uniku.ac.id)

# Mahasiswa mampu

- Memahami Metodologi Pengembangan Sistem
- Dapat mengidentifikasi Proyek Sistem
- Dapat menginisialisasi dan merencanakan proyek
- Memahami tahapan-tahapan dalam pengembangan sistem

# Analisis dan Perancangan Sistem

1. Metodologi pengembangan Sistem
2. Identifikasi dan seleksi proyek
3. Inisiasi dan perencanaan proyek
4. Tahapan Analisa
5. Tahapan Desain
6. Implementasi
7. Maintenances
8. Kelemahan dari SDLC tradisional

# **Metodologi pengembangan Sistem**

**Proses-proses standard** yang digunakan untuk membangun suatu sistem informasi meliputi langkah-langkah :

- Analisa
- Desain
- Implementasi
- Maintenance

Pada perkembangannya, proses-proses standar tadi dituangkan dalam satu metode yang dikenal dengan nama Systems Development Life Cycle (SDLC) atau siklus hidup pengembangan sistem yang merupakan metodologi umum dalam pengembangan sistem yang menandai kemajuan dari usaha analisa dan desain.

## **Fase-fase dalam SDLC**

1. Identifikasi dan seleksi proyek
2. Inisiasi dan perencanaan proyek
3. Analisa
4. Desain
  - o Desain logikal
  - o Desain Fisikal
5. Implementasi
6. Maintenance

# 1. Identifikasi dan seleksi proyek

Langkah pertama dalam SDLC

Keseluruhan informasi yang dibutuhkan oleh sistem **diidentifikasi, dianalisa, diprioritaskan dan disusun ulang.**

Beberapa hal yang dilakukan :

- o Mengidentifikasi proyek-proyek yang potensial
- o Melakukan klasifikasi dan me-rangking proyek
- o Memilih proyek untuk dikembangkan.

# 1. Identifikasi dan seleksi proyek

Sumber daya yang terlibat,



*user,*



*sistem analis*



*manager yang  
mengkoordinasi proyek*

# 1. Identifikasi dan seleksi proyek

Aktivitas :

- *mewawancarai manajemen user*
- *merangkum pengetahuan yang didapatkan,*
- *Mengestimasi cakupan proyek*
- *mendokumentasikan hasilnya*

Output :

Laporan kelayakan berisi definisi masalah dan rangkuman tujuan yang ingin dicapai

## **2. Inisiasi dan perencanaan proyek**

Dalam tahapan ini Proyek SI yang potensial dijelaskan dan argumentasi dikemukakan untuk melanjutkan proyek.

**Hasil dari tahapan ini :**

- Langkah-langkah detail
- Rencana kerja
- High level system requirement
- Penugasan untuk anggota tim.

### 3. Tahapan Analisa

Fase ketiga dalam SDLC,

- *Mempelajari Sistem Yang Sedang Berjalan*
- *Mengusulkan sistem pengganti.*

Dalam tahapan ini

- **dideskripsikan** sistem yang sedang berjalan,
- **masalah dan kesempatan didefinisikan**, dan
- **rekomendasi** umum untuk bagaimana memperbaiki, meningkatkan atau mengganti sistem yang sedang berjalan diusulkan.

Tujuan utama :

memahami dan mendokumentasikan kebutuhan bisnis (business need) dan persyaratan proses dari sistem baru.

### **3. Tahapan Analisa**

#### **6 aktifitas utama**

1. Pengumpulan informasi
2. Mendefinisikan sistem kebutuhan (requirement)
3. Membangun prototype untuk menemukan kebutuhan
4. Memprioritaskan Kebutuhan
5. Menyusun dan mengevaluasi alternatif
6. Mereview kebutuhan dengan pihak manajemen

## 4. Tahapan Desain

Pada tahapan ini deskripsi dari kebutuhan yang telah direkomendasikan **diubah** ke dalam spesifikasi sistem **physical** dan **logical**.

## Logical Design

Semua fitur-fitur fungsional dari sistem dipilih dari tahapan analisis dideskripsikan terpisah dari platform komputer yang nanti digunakan.

Hasil dari tahapan ini :

- Deskripsi fungsional : data dan proses → dalam sistem baru
- Deskripsi yang detail dari spesifikasi sistem meliputi:
  - Input
  - Output
  - Process

## **Physical design**

Pada bagian ini spesifikasi logical diubah ke dalam detail teknologi dimana pemrograman dan pengembangan sistem bisa diselesaikan.

Output dari sistem ini :

- Deskripsi teknikal
- Deskripsi yang detail dari spesifikasi sistem meliputi:
  - programs,
  - files,
  - network,
  - system software

Aktifitas utama pada tahapan desain;

- o Merancang dan mengintegrasikan network
- o Merancang Arsitektur aplikasi
- o Mendesain user interface
- o Mendesain sistem interface
- o Mendesain dan mengintegrasikan database
- o Memuat prototype untuk detail dari desain
- o Mendesain dan mengintegrasikan kendali sistem

## 5. Implementasi

Tahapan kelima yaitu:

- Coding
- Testing
- Installasi

Output dari tahapan ini adalah :

- source code
- prosedur
- pelatihan.

## 6. Maintenance

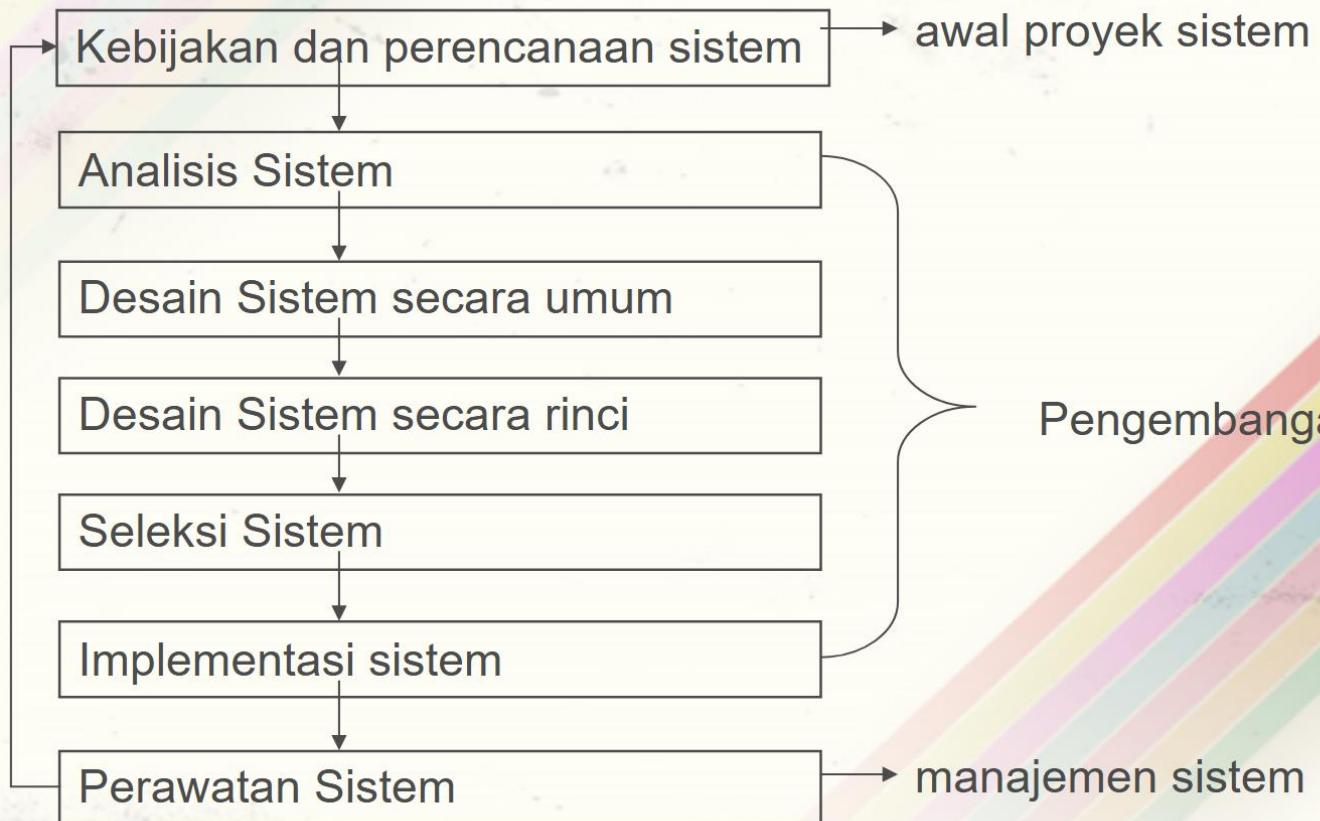
Langkah terakhir dari SDLC dimana pada tahapan ini sistem secara sistematis diperbaiki dan ditingkatkan.

Hasil dari tahapan ini adalah  
Versi baru dari software yang telah dibuat.

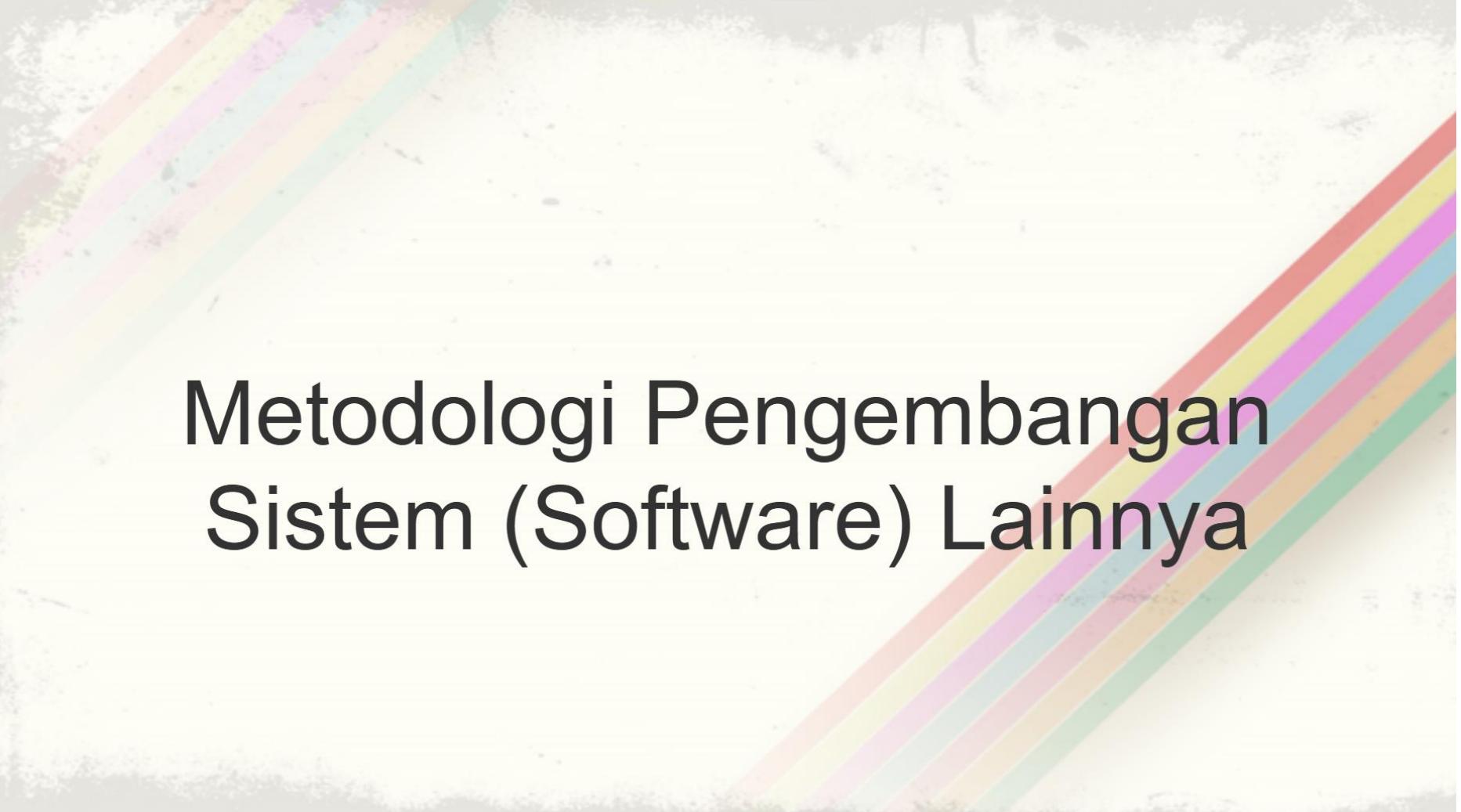
# Kelemahan dari SDLC tradisional

- Terlalu mahal (biaya dan waktu) ketika terjadi perubahan, sistem sudah dikembangkan
- SDLC merupakan metode dengan **pendekatan terstruktur** yang mensyaratkan mengikuti semua langkah yang ada.
- Biaya maintenance cukup besar

# SIKLUS HIDUP PENGEMBANGAN SISTEM



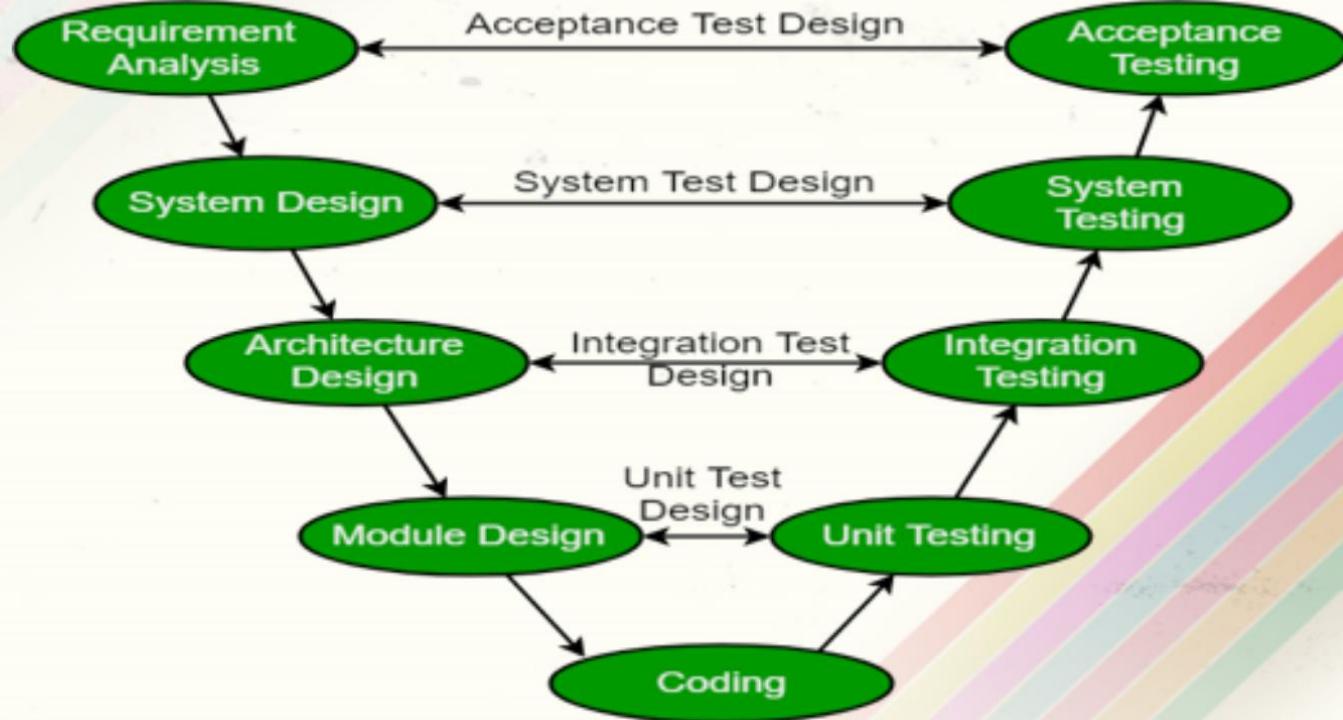
Jogiyanto HM, Analisis dan Perancangan Sistem Informasi



# Metodologi Pengembangan Sistem (Software) Lainnya

## SDLC V-Model

- Model V adalah jenis model SDLC dimana proses dijalankan secara berurutan dalam bentuk V. Dikenal sebagai model Verifikasi dan Validasi.
- Pengembangan setiap langkah terkait langsung dengan tahap pengujian. Fase berikutnya dimulai hanya setelah menyelesaikan fase sebelumnya yaitu untuk setiap aktivitas pengembangan, ada aktivitas pengujian yang sesuai dengannya.



SDLC V-Model

## **Design Phase:**

- **Requirement Analysis:**

Fase ini berisi komunikasi terperinci dengan pelanggan untuk memahami persyaratan dan harapan mereka. Tahap ini dikenal sebagai Requirement Gathering.

- **System Design:**

Fase ini berisi desain sistem dan perangkat keras lengkap dan pengaturan komunikasi untuk mengembangkan produk.

- **Architectural Design:**

Desain sistem dipecah lebih jauh menjadi modul-modul yang menggunakan berbagai fungsi. Transfer data dan komunikasi antara modul internal dan dengan dunia luar (sistem lain) dipahami dengan jelas

- **Module Design:**

Dalam fase ini sistem dipecah menjadi modul-modul kecil. Desain rinci modul ditentukan, juga dikenal sebagai Low-Level Design (LLD)

## **Testing Phases:**

- **Unit Testing:**

Rencana Uji Unit dikembangkan selama fase desain modul.

Rencana Uji Unit ini dijalankan untuk menghilangkan bug pada kode atau tingkat unit.

- **Integration testing:**

Setelah menyelesaikan pengujian unit, pengujian integrasi dilakukan. Dalam pengujian integrasi, modul diintegrasikan dan sistem diuji. Pengujian integrasi dilakukan pada tahap desain Arsitektur. Tes ini memverifikasi komunikasi modul di antara mereka sendiri.

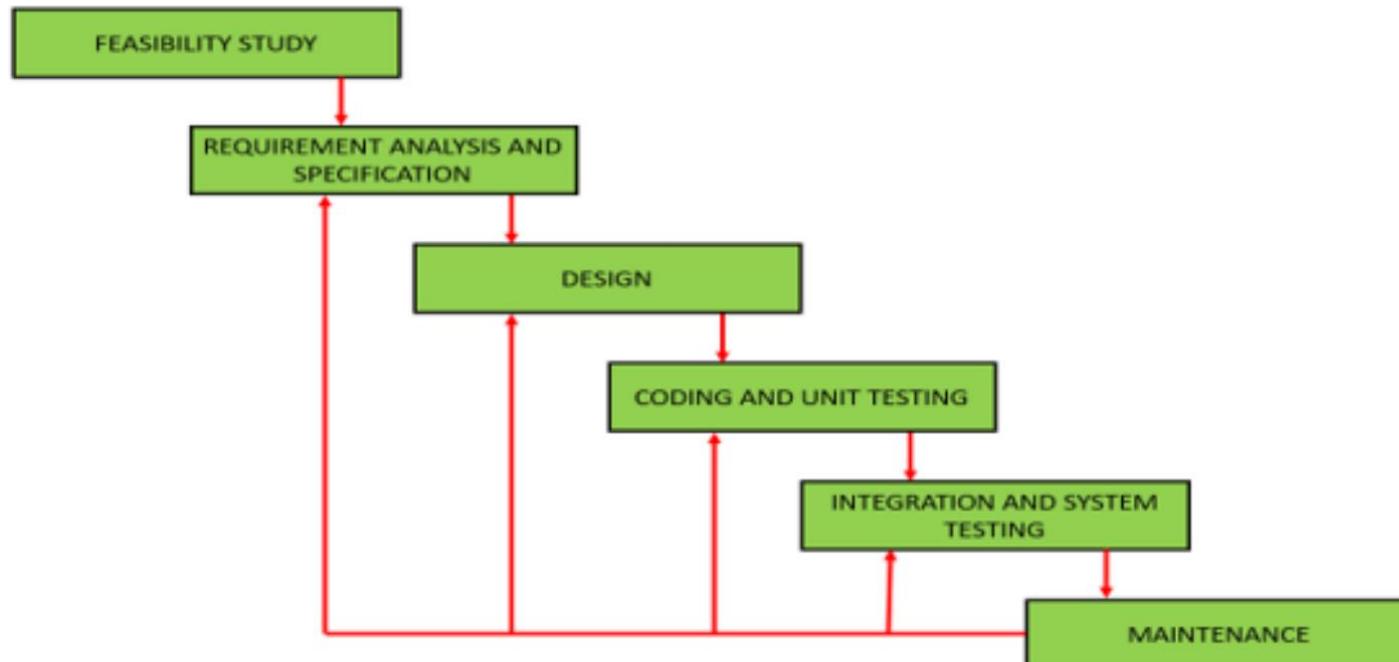
- **System Testing**

Pengujian sistem menguji aplikasi lengkap dengan fungsionalitas, antar ketergantungan, dan komunikasinya, menguji persyaratan fungsional dan non-fungsional dari aplikasi yang dikembangkan.

- **User Acceptance Testing (UAT)**

UAT dilakukan di lingkungan pengguna yang menyerupai lingkungan produksi. UAT memverifikasi bahwa sistem yang dikirimkan memenuhi persyaratan pengguna dan sistem siap digunakan di dunia nyata.

# Iterative Waterfall Model



- **Feasibility Study:** Tujuan utama dari tahap ini adalah untuk menentukan kelayakan secara finansial dan teknis untuk mengembangkan perangkat lunak. Studi kelayakan melibatkan pemahaman masalah dan kemudian menentukan berbagai kemungkinan strategi untuk menyelesaikan masalah. Solusi teridentifikasi yang berbeda ini dianalisis berdasarkan manfaat dan kekurangannya, Solusi terbaik dipilih dan semua fase lainnya dilakukan sesuai dengan strategi solusi ini.

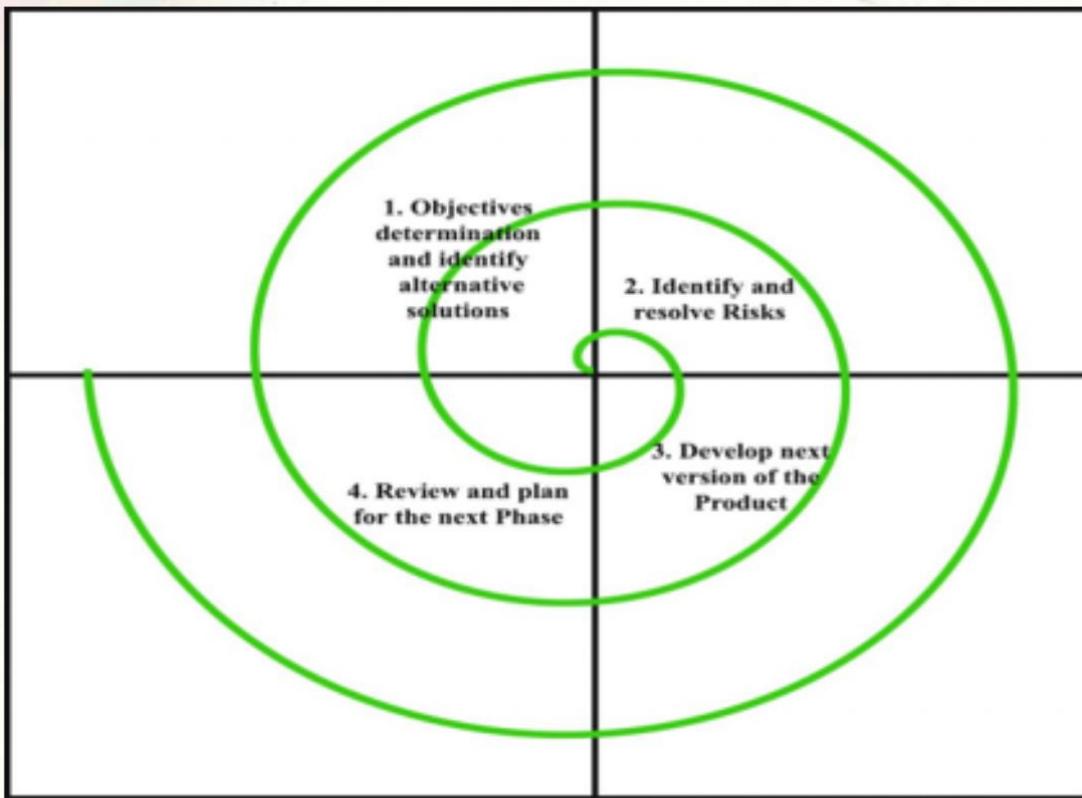
- **Requirements analysis and specification:** Tujuan dari tahap analisis dan spesifikasi kebutuhan adalah untuk memahami persyaratan yang tepat dari pelanggan dan mendokumentasikannya dengan benar. Fase ini terdiri dari dua aktivitas berbeda yaitu **Requirement gathering and analysis, Requirement specification**

- **Requirements analysis and specification:**
  - **Requirement gathering and analysis:** Pertama, semua persyaratan terkait perangkat lunak dikumpulkan dari pelanggan dan kemudian persyaratan yang dikumpulkan dianalisis. Tujuan dari bagian analisis adalah untuk menghilangkan ketidaklengkapan (persyaratan yang tidak lengkap adalah persyaratan di mana beberapa bagian dari persyaratan aktual telah dihilangkan) dan ketidakkonsistenan (persyaratan yang tidak konsisten adalah persyaratan di mana beberapa bagian persyaratan bertentangan dengan bagian lain).
  - **Requirement specification:** Persyaratan yang dianalisis ini didokumentasikan dalam dokumen spesifikasi kebutuhan perangkat lunak (SRS). Dokumen SRS berfungsi sebagai kontrak antara tim pengembangan dan pelanggan. Setiap perselisihan di masa depan antara pelanggan dan pengembang dapat diselesaikan dengan memeriksa dokumen SRS.

- **Design:** Tujuan dari tahap desain adalah untuk mengubah persyaratan yang ditentukan dalam dokumen SRS menjadi struktur yang sesuai untuk diimplementasikan dalam beberapa bahasa pemrograman
- **Coding and Unit testing:** Dalam tahap pengkodean desain perangkat lunak diterjemahkan ke dalam kode sumber menggunakan bahasa pemrograman yang sesuai. Jadi setiap modul yang dirancang diberi kode. Tujuan dari tahap pengujian unit adalah untuk memeriksa apakah setiap modul berfungsi dengan baik atau tidak.

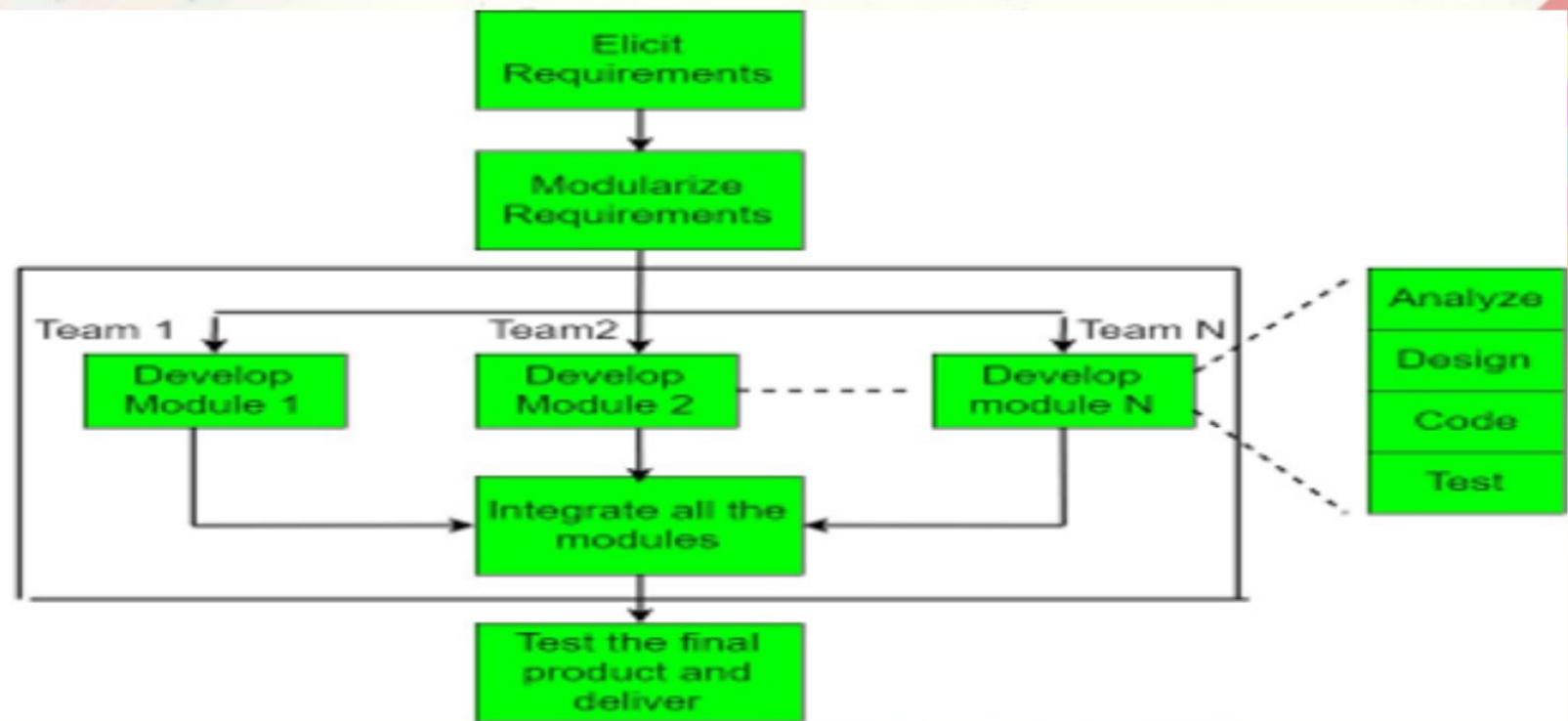
# Spiral Model

- Model spiral adalah salah satu model Siklus Hidup Pengembangan Perangkat Lunak yang paling penting, yang memberikan dukungan untuk Penanganan Risiko.
- Dalam representasi diagramnya, terlihat seperti spiral dengan banyak loop. Jumlah pasti dari loop spiral tidak diketahui dan dapat bervariasi dari satu proyek ke proyek lainnya. Setiap lingkaran spiral disebut Fase proses pengembangan perangkat lunak. Jumlah fase yang dibutuhkan untuk mengembangkan produk dapat bervariasi oleh manajer proyek tergantung pada risiko proyek.
- Radius spiral di titik mana pun mewakili biaya (biaya) proyek, dan dimensi sudut mewakili kemajuan yang dibuat dalam fase saat ini.



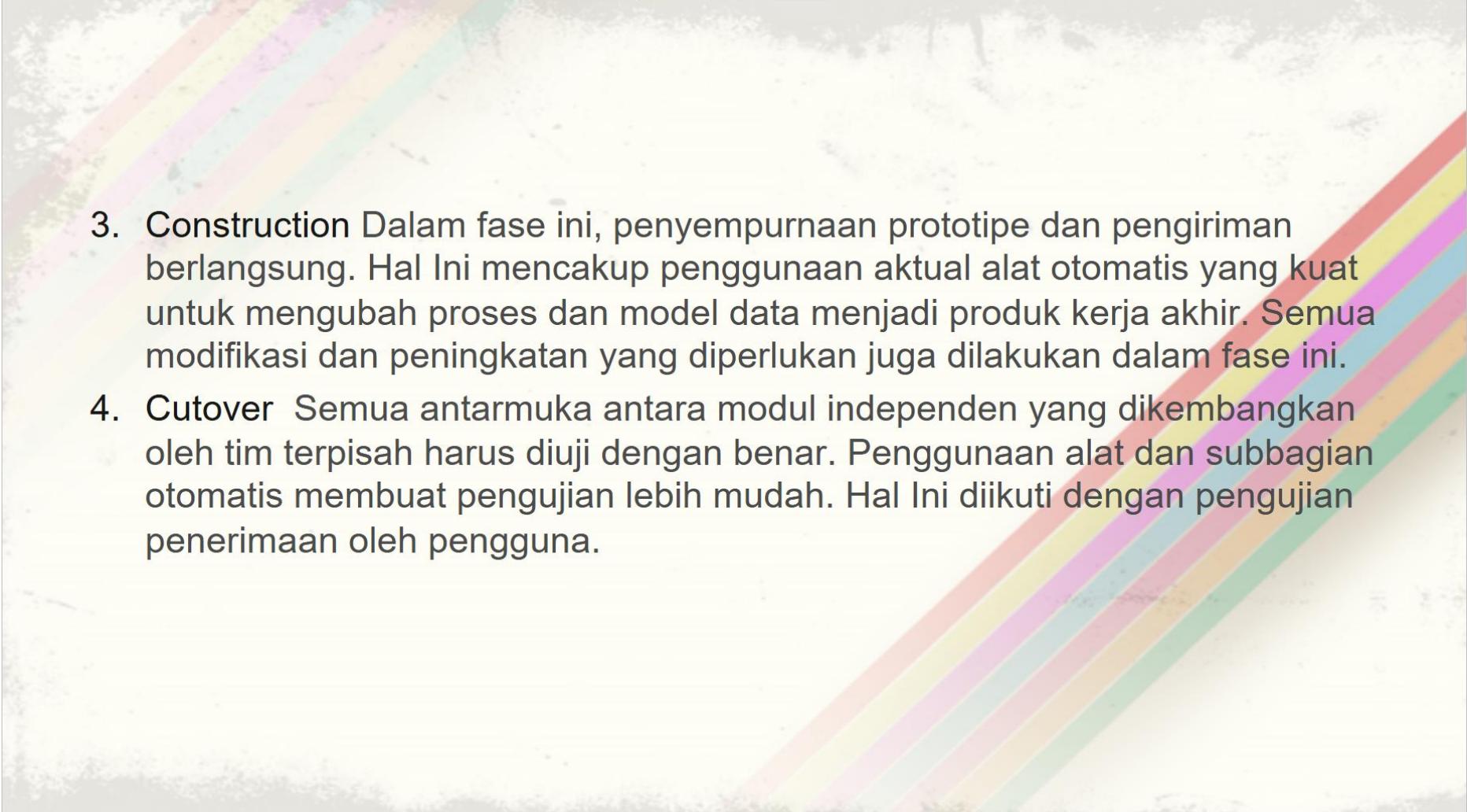
- Setiap phase pada model spiral dibagi dalam 4 kuadran
1. **Objectives determination and identify alternative solutions:** Persyaratan dikumpulkan dari pelanggan dan tujuan diidentifikasi, diuraikan, dan dianalisis pada awal setiap fase. Kemudian solusi alternatif yang memungkinkan untuk fase tersebut diusulkan di kuadran ini.
  2. **Identify and resolve Risks:** Selama kuadran kedua, semua solusi yang mungkin dievaluasi untuk memilih solusi terbaik. Kemudian risiko yang terkait dengan solusi tersebut diidentifikasi dan risiko tersebut diselesaikan dengan menggunakan strategi terbaik. Di akhir kuadran ini, Prototipe dibangun untuk solusi terbaik.
  3. **Develop next version of the Product:** Selama kuadran ketiga, fitur yang diidentifikasi dikembangkan dan diverifikasi melalui pengujian. Di akhir kuadran ketiga, versi perangkat lunak berikutnya tersedia.
  4. **Review and plan for the next Phase:** Di kuadran keempat, Pelanggan mengevaluasi versi perangkat lunak yang dikembangkan sejauh ini. Pada akhirnya, perencanaan untuk tahap selanjutnya dimulai.

# Rapid application development model (RAD)

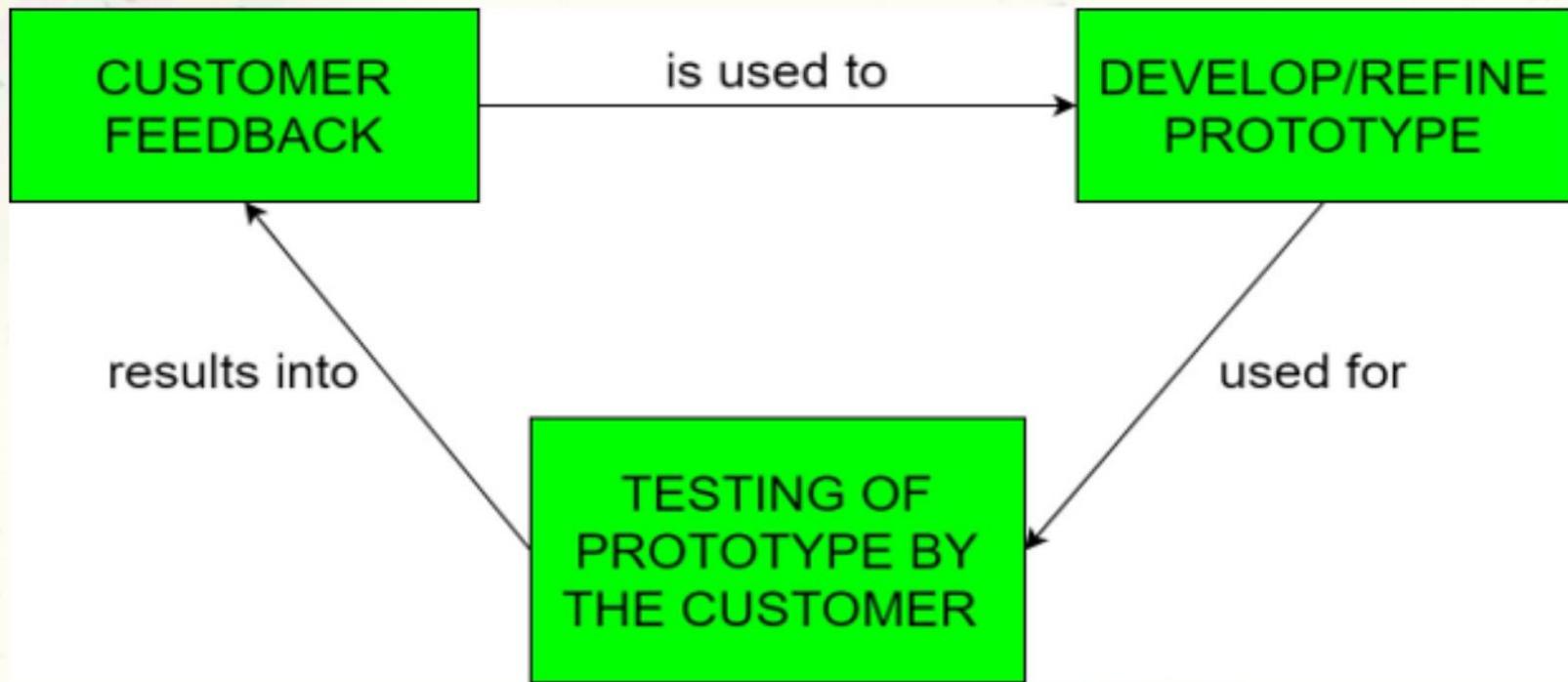


## 4 basic phases:

1. Requirements Planning Melibatkan penggunaan berbagai teknik yang digunakan dalam pengumpulan persyaratan seperti brainstorming, analisis tugas, analisis formulir, skenario pengguna, FAST (Teknik Pengembangan Aplikasi yang Difasilitasi), dll. juga terdiri dari seluruh rencana terstruktur yang menggambarkan data penting, metode untuk mendapatkannya dan kemudian memprosesnya untuk membentuk model akhir yang disempurnakan.
2. User Description Fase ini mengambil umpan balik pengguna dan membangun prototipe menggunakan alat pengembang. hal ini mencakup pemeriksaan ulang dan validasi data yang dikumpulkan pada tahap pertama. Atribut set data juga diidentifikasi dan dijelaskan dalam fase ini.

- 
3. Construction Dalam fase ini, penyempurnaan prototipe dan pengiriman berlangsung. Hal Ini mencakup penggunaan aktual alat otomatis yang kuat untuk mengubah proses dan model data menjadi produk kerja akhir. Semua modifikasi dan peningkatan yang diperlukan juga dilakukan dalam fase ini.
  4. Cutover Semua antarmuka antara modul independen yang dikembangkan oleh tim terpisah harus diuji dengan benar. Penggunaan alat dan subbagian otomatis membuat pengujian lebih mudah. Hal Ini diikuti dengan pengujian penerimaan oleh pengguna.

# Prototyping Model



- Dalam model ini, sistem diimplementasikan secara parsial sebelum atau selama tahap analisis sehingga memberikan kesempatan kepada pelanggan untuk melihat produk di awal siklus hidup.
- Prosesnya dimulai dengan mewawancara pelanggan dan mengembangkan model kertas tingkat tinggi yang tidak lengkap.
- Dokumen ini digunakan untuk membangun prototipe awal yang hanya mendukung fungsionalitas dasar yang diinginkan oleh pelanggan.
- Setelah pelanggan mengetahui masalahnya, prototipe tersebut selanjutnya disempurnakan untuk menghilangkannya.
- Proses berlanjut sampai pengguna menyetujui prototipe dan menemukan model kerja yang memuaskan.



Terima Kasih