

# REKAYASA PERANGKAT LUNAK

OLEH : RAHMAT ROBI WALIYANSYAH, M.KOM.

# Curriculum vitae

1. Nama : Rahmat Robi Waliyansyah, M.Kom.
2. TTL : Jambi, 25 Oktober 1988
3. Agama : Islam
4. Status : Menikah
5. Alamat : Jl. Sawah Besar XI No.05 RT.02 RW.06 Kel. Kaligawe Kec. Gayamsari, Kota Semarang 50164
6. E-Mail : [rahmat.robi.waliyansyah@gmail.com](mailto:rahmat.robi.waliyansyah@gmail.com)
7. Phone : +6285377479974 (WA)
8. Pendidikan :
  - S1 = Universitas Putra Indonesia-YPTK, Padang
  - S2 = Universitas Diponegoro, Semarang



# KONTRAK PERKULIAHAN

- Nama Matakuliah : Rekayasa Perangkat Lunak
- Semester : V / Lima
- SKS : 2 SKS

# Deskripsi Perkuliahan

- ▶ Mata kuliah ini mengkaji konsep Rekayasa Perangkat Lunak dan menerapkannya dalam membangun produk perangkat lunak yang benar.

# Tujuan

1. Mahasiswa memahami metode-metode rekayasa perangkat lunak
2. Mahasiswa memahami proses membangun perangkat lunak yang efektif dan efisien
3. Mahasiswa mampu menggunakan metode rekayasa perangkat lunak berorientasi objek



# STRATEGI PERKULIAHAN

- Kuliah diberikan kepada mahasiswa SI Informatika UPGRIS yang mengambil matakuliah ini sebagai pilihan. Perkuliahan dilakukan sebanyak 14 kali pertemuan kuliah tatap muka. Metode perkuliahan adalah kombinasi antara ceramah, diskusi, dan diakhiri dengan presentasi tugas kelompok/proyek akhir.
- Mahasiswa wajib mengikuti perkuliahan minimal 80 persen, dan presentasi proyek akhir 100 persen. Mahasiswa pengulang matakuliah Rekayasa Perangkat Lunak diwajibkan mengikuti keseluruhan kegiatan kuliah dan presentasi tugas kelompok/proyek akhir selama satu semester.

# Tugas

---

- Tugas terdiri dari dua jenis, yaitu **tugas perorangan** yang harus diselesaikan oleh mahasiswa pada waktu tertentu, dan **tugas kelompok** terdiri atas beberapa orang. Topik yang dipilih adalah bebas, dengan syarat tidak ada yang sama di antara kelompok. Produk yang dihasilkan oleh setiap kelompok bisa berupa program komputer, laporan hasil kajian sesuai dengan topik yang dipilih, dan slide presentasi.

# Referensi

- I. Roger S. Pressman, Software Engineering A Practioner's Apporach, 2014
- II. Ian Sommerville, Software Engineering (10th Edition), 2015



# KRITERIA PENILAIAN

- Nilai akhir (NA) adalah nilai kumulatif dari nilai ujian tengah semester (UTS), ujian akhir semester (UAS), tugas perorangan (TP), dan tugas kelompok atau proyek akhir (PA). Metode dan bobot nilai sebagai berikut:
  1. UTS (1-7) dan UAS (9-15) dilakukan melalui ujian tertulis dengan bobot masing-masing 35%. Kisi-kisi ujian akan disampaikan pada pertemuan ke-7 untuk UTS, dan pada pertemuan ke-15 untuk UAS.
  2. Nilai TP adalah rata-rata dari semua tugas yang diberikan, dan diberi bobot 10%
  3. Nilai PA terdiri dari nilai produk proyek (program komputer, laporan) dan presentasi. Bobot nilai PA adalah 20%.

- ▶ Kuliah dilaksanakan pada setiap hari Rabu pukul 13:00-14:40 di Ruang GP. 608
- ▶ Keterlambatan maksimal 15 Menit.

## JADWAL KULIAH

# OUTLINE

---

1. Pengantar RPL
2. Software Development Life Cycle (SDLC)
3. Rekayasa Sistem
4. Rekayasa Kebutuhan
5. Analisis Pengembangan Perangkat Lunak
6. Konsep Perancangan Perangkat Lunak
7. Perancangan Arsitektur
8. UTS

# OUTLINE

9. Pendahuluan Rekayasa Perangkat Lunak Berorientasi Objek (OOSE)
10. Model Kebutuhan OOSE
11. Model Analisis OOSE
12. Konstruksi OOSE bagian 1
13. Konstruksi OOSE bagian 2
14. Implementasi OOSE
15. Studi Kasus OOSE
16. UAS

# Definisi perangkat lunak Definisi Perangkat lunak (software)

- Definisi Perangkat lunak (software) adalah istilah khusus untuk data yang diformat, dan disimpan **secara digital**, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca, dan ditulis oleh komputer. Dengan kata lain, bagian sistem komputer yang tidak berwujud.

# **PROBLEM DALAM PEMBUATAN PERANGKAT LUNAK**

- Tidak memiliki waktu yang cukup dalam mengumpulkan data pada proses pembuatan perangkat lunak.
- Ketidakpuasan user pada Software yang dibuat.
- Kualitas Software terkadang meragukan.
- Sulit dalam maintenance Software sekarang.



# Rekayasa perangkat lunak

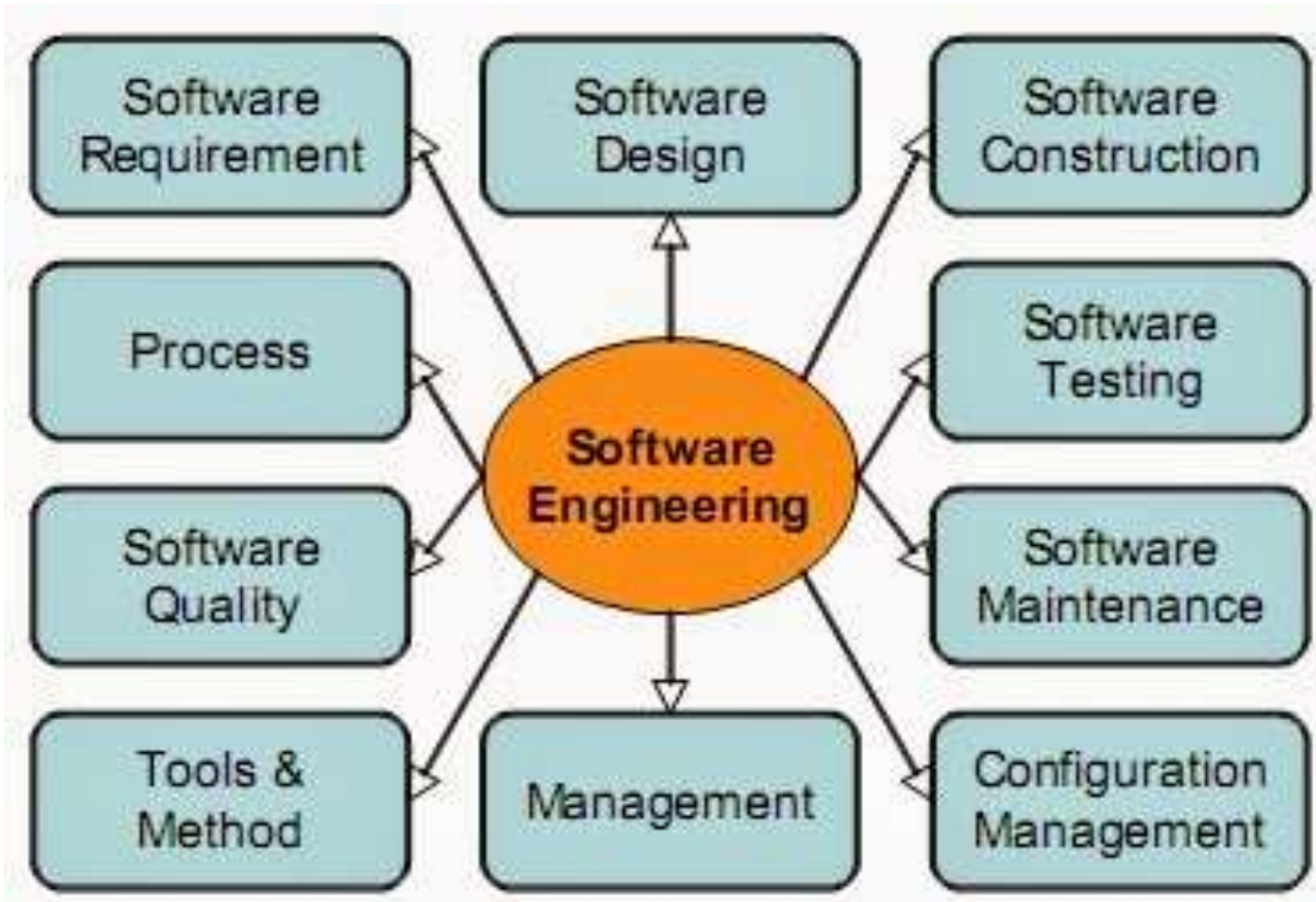
- Rekayasa perangkat lunak adalah satu bidang profesi yang mendalami cara-cara pengembangan perangkat lunak termasuk pembuatan, pemeliharaan, manajemen organisasi pengembangannya perangkat lunak dan manajemen kualitas.

# Tujuan Rekayasa Perangkat Lunak

- ❖Memperoleh biaya produksi perangkat lunak yang rendah.
- ❖Menghasilkan perangkat lunak yang kinerjanya tinggi, andal dan tepat waktu.
- ❖Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis platform.
- ❖Menghasilkan perangkat lunak yang biaya perawatannya rendah.

# Rekayasa Perangkat Lunak dan Disiplin Ilmu Lain





## **RUANG LINGKUP RPL**

Sesuai dengan definisi yang telah disampaikan sebelumnya, maka ruang lingkup RPL dapat digambarkan sebagai berikut:

# RUANG LINGKUP RPL

---

- **Software Requirements** berhubungan dengan spesifikasi kebutuhan dan persyaratan perangkat lunak
- **Software Desain** mencakup proses penampilan arsitektur, komponen, antar muka, dan karakteristik lain dari perangkat lunak
- **Software Construction** berhubungan dengan detail pengembangan perangkat lunak, termasuk algoritma, pengkodean, pengujian dan pencarian kesalahan
- **Software Testing** meliputi pengujian pada keseluruhan perilaku perangkat lunak
- **Software Maintenance** mencakup upaya-upaya perawatan ketika perangkat lunak telah dioperasikan



# *RUANG LINGKUP RPL*

- **Software Configuration Management** berhubungan dengan usaha perubahan konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu
  - **Software Engineering management** berkaitan dengan pengelolaan dan pengukuran RPL, termasuk perencanaan proyek perangkat lunak
  - **Software Engineering Tools And Methods** mencakup kajian teoritis tentang alat bantu dan metode RPL
  - **Software Engineering Process** berhubungan dengan definisi, implementasi pengukuran, pengelolaan, perubahan dan perbaikan proses RPL
  - **Software Quality** menitik beratkan pada kualitas dan daur hidup perangkat lunak
- 



# Kriteria Dalam Merekayasa Perangkat Lunak

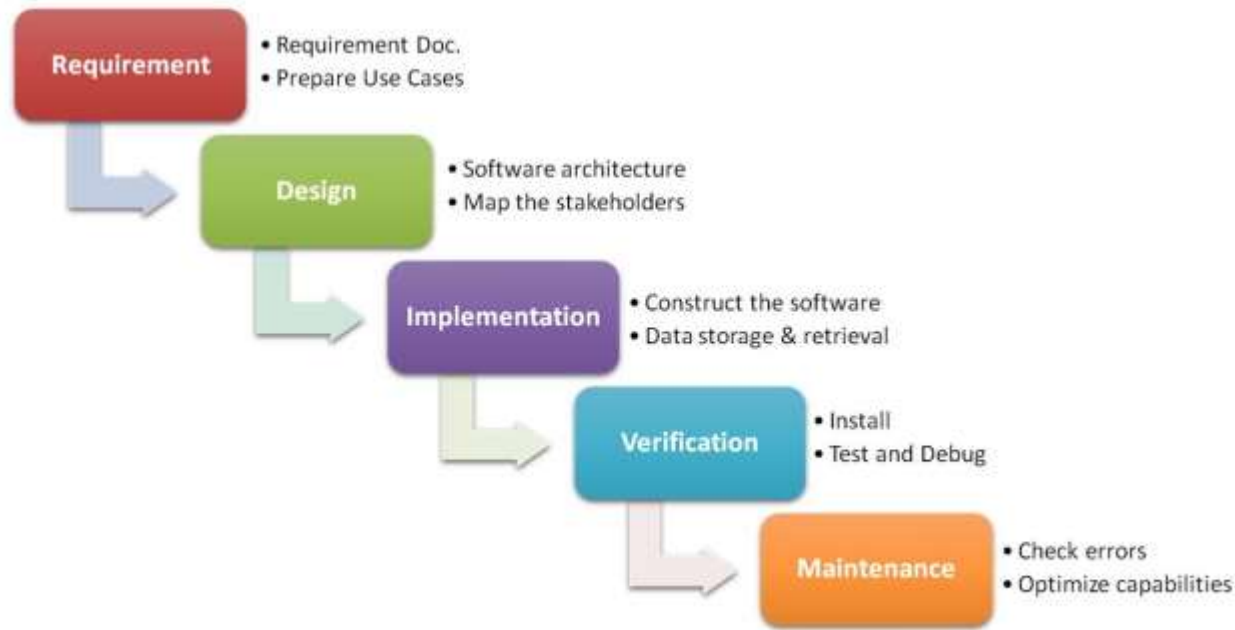
- ❑ Dapat terus dirawat dan dipelihara (maintainability).
- ❑ Dapat mengikuti perkembangan teknologi (dependability).
- ❑ Dapat mengikuti keinginan pengguna (robust)..
- ❑ Efektif dan efisien dalam menggunakan energi dan penggunaannya.
- ❑ Dapat memenuhi kebutuhan yang diinginkan (usability).

# Model Model Proses Pengembangan Rekayasa Perangkat Lunak

- ▶ Model Sekuensial Linier atau Waterfall Development Model
- ▶ Model Prototype
- ▶ Model Rapid Application Development (RAD)
- ▶ Model Evolutionary Development / Evolutionary Software Process Models
  1. Model Incremental
  2. Model Spiral / Model Boehm

# Model Sekuensial Linier atau Waterfall Development Model

---



Model Sekuensial Linier atau sering disebut Model Pengembangan Air Terjun, merupakan paradigma model pengembangan perangkat lunak paling tua, dan paling banyak dipakai. Model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat analisis, desain, kode, pengujian, dan pemeliharaan.

# Model Sekuensial Linier atau Waterfall Development Model

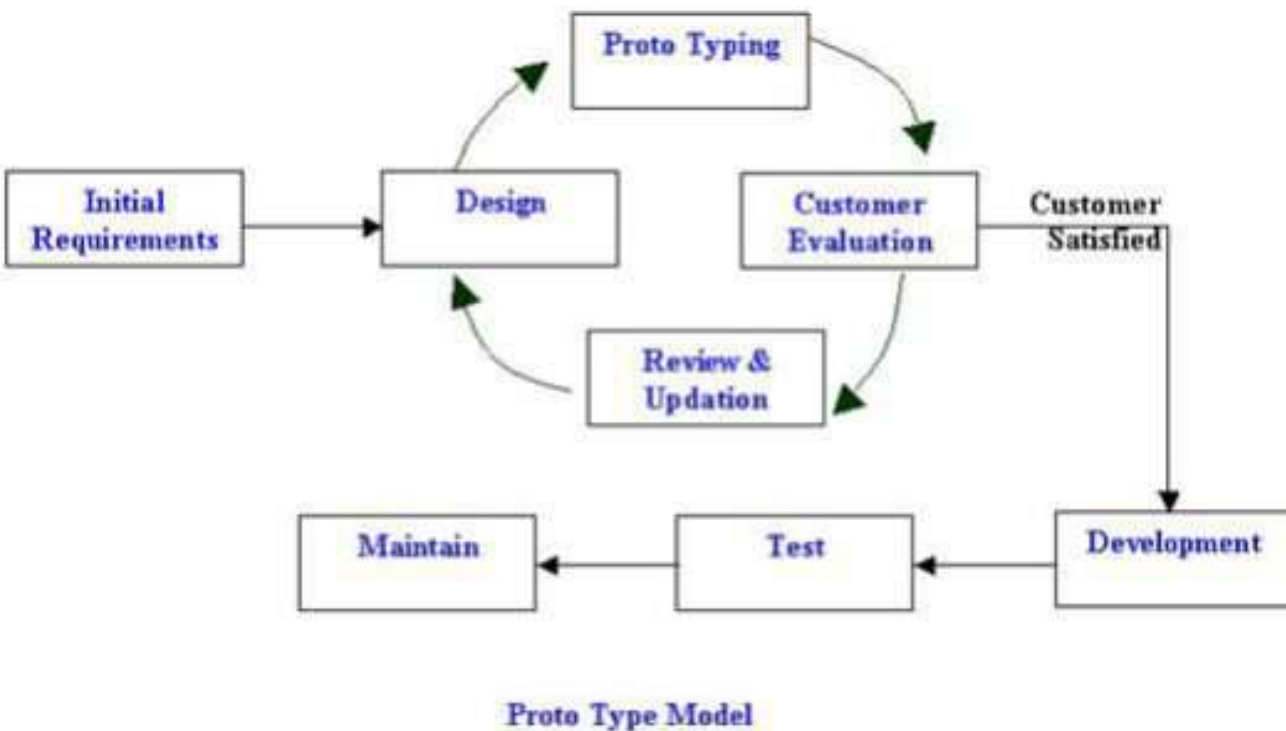
## Kekurangan Model Waterfall:

- ❑ Terjadinya pembagian proyek menjadi tahap-tahap yang tidak fleksibel, karena komitmen harus dilakukan pada tahap awal proses.
- ❑ Model air terjun harus digunakan hanya ketika persyaratan dipahami dengan baik.

## Kelebihan Model Waterfall:

- ❑ Bisa digunakan jika suatu persyaratan untuk membuat suatu software sudah dipahami dengan baik dan sudah lengkap semua persyaratan yang ada.

# Model Prototype



- Dalam Model Prototype, prototype dari perangkat lunak yang dihasilkan kemudian dipresentasikan kepada pelanggan, dan pelanggan tersebut diberikan kesempatan untuk memberikan masukan sehingga perangkat lunak yang dihasilkan nantinya betul-betul sesuai dengan keinginan dan kebutuhan pelanggan.
- Perubahan dan presentasi prototype dapat dilakukan berkali-kali sampai dicapai kesepakatan bentuk dari perangkat lunak yang akan dikembangkan.



# Model Prototype

## Keunggulan Prototyping:

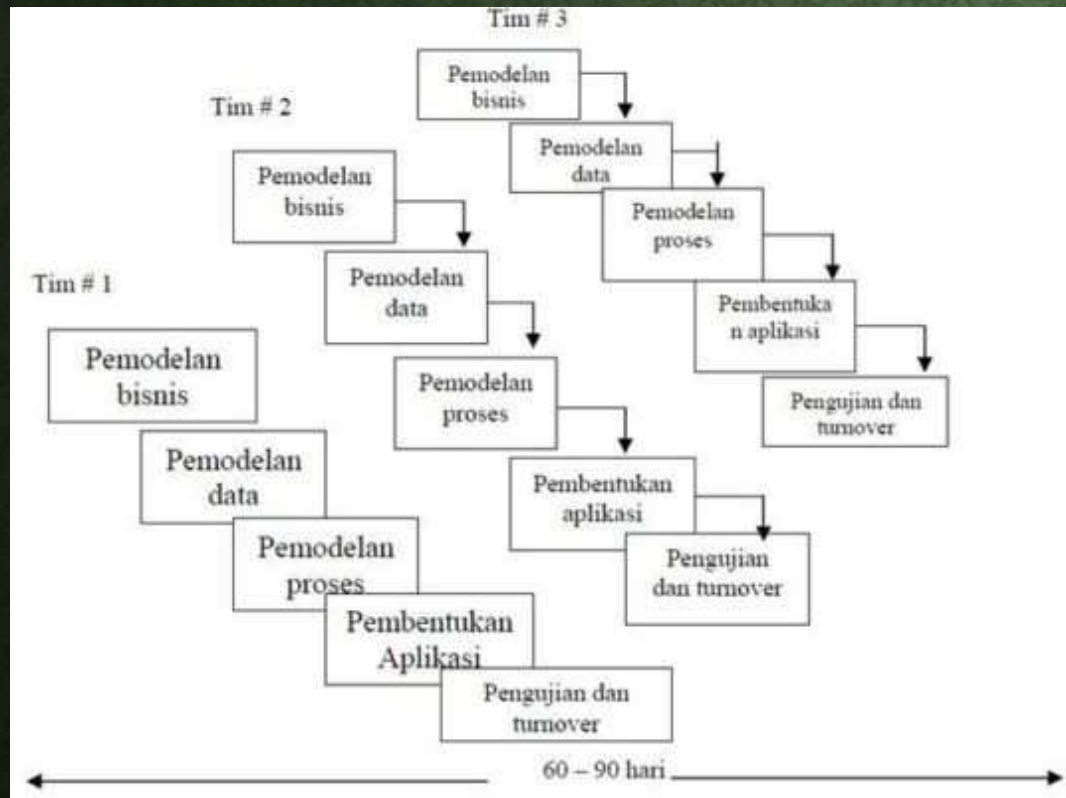
- User dapat berpartisipasi aktif
- Penentuan kebutuhan lebih mudah diwujudkan
- Mempersingkat waktu pengembangan SI

## Kelemahan Prototyping :

- Proses analisis dan perancangan terlalu singkat
- Mengesampingkan alternatif pemecahan masalah
- Biasanya kurang fleksible dalam menghadapi perubahan
- Prototype yang dihasilkan tidak selamanya mudah dirubah
- Prototype terlalu cepat selesai



# MODEL RAPID APPLICATION DEVELOPMENT (RAD)



- Rapid Application Model (RAD) adalah sebuah proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek.
- Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen.
- Jika kebutuhan dipahami dengan baik, proses RAD memungkinkan tim pengembangan menciptakan “sistem fungsional yang utuh” dalam periode waktu yang sangat pendek (kira-kira 60 sampai 90 hari).

# MODEL RAPID APPLICATION DEVELOPMENT (RAD)

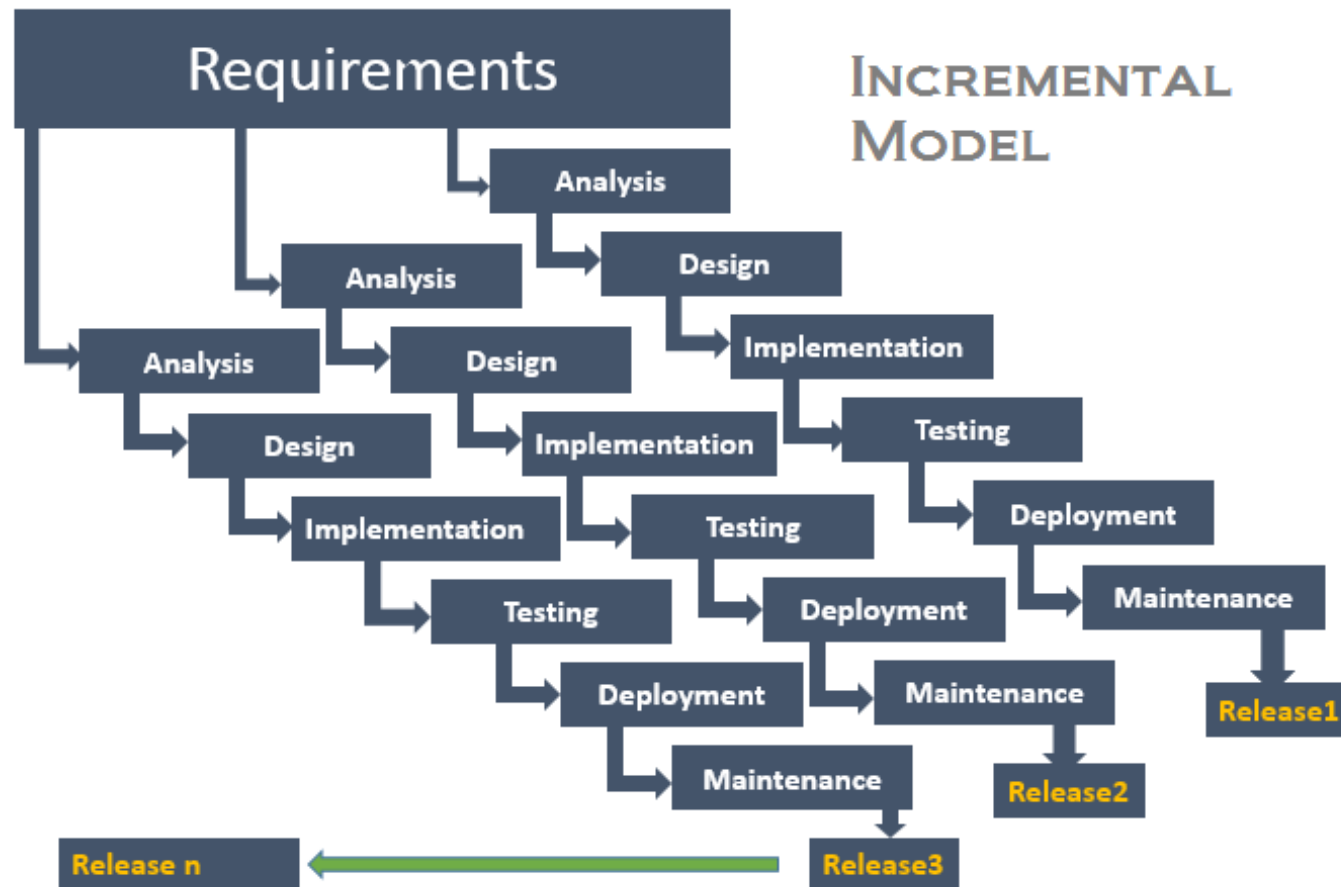
## Kelebihan Penggunaan Model RAD

- Dimungkinkan dalam proses pembuatan membutuhkan waktu yang sangat singkat (60-90 hari).
- Menghemat biaya, karena penekannya adalah penggunaan komponen-komponen yang sudah ada.
- RAD menggunakan kembali komponen-komponen yang sudah ada, maka beberapa komponen program sudah diuji sehingga kita dapat melakukan penghematan waktu dalam uji coba

## Kekurangan Penggunaan Model RAD

- RAD tidak cocok digunakan untuk sistem yang mempunyai resiko teknik yang tinggi.
- Membutuhkan Tenaga kerja yang banyak untuk menyelesaikan sebuah proyek dalam skala besar.
- Jika ada perubahan di tengah-tengah pengerjaan maka harus membuat kontrak baru antara pengembang dan pelanggan.

# Model Incremental



Model Incremental merupakan hasil kombinasi elemen-elemen dari model waterfall yang diaplikasikan secara berulang, atau bisa disebut gabungan dari Model linear sekuensial (waterfall) dengan Model Prototype. Elemen-elemen tersebut dikerjakan hingga menghasilkan produk dengan spesifikasi tertentu kemudian proses dimulai dari awal kembali hingga muncul hasil yang spesifikasinya lebih lengkap dari sebelumnya dan tentunya memenuhi kebutuhan pemakai.



# Model Incremental

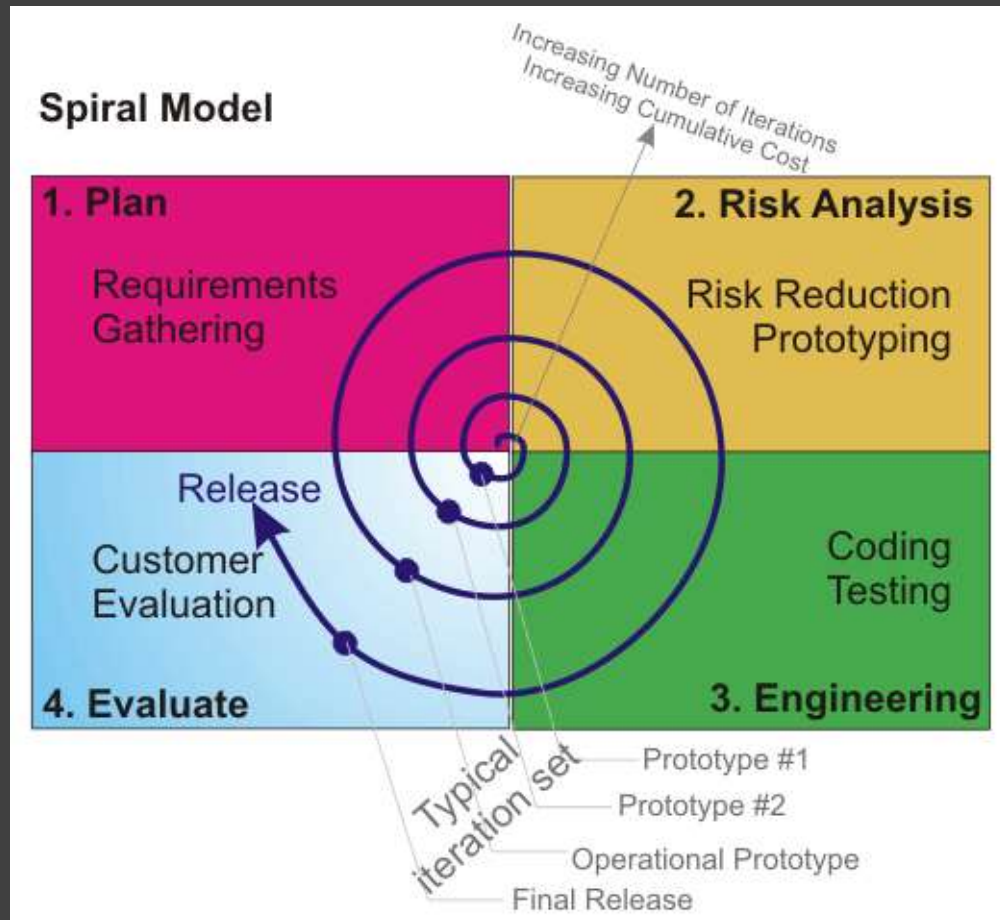
## Keunggulan dari Incremental Model :

- Personil bekerja optimal
- Pihak konsumen dapat langsung menggunakan dahulu bagian-bagian yang telah selesai dibangun. Contohnya pemasukan data karyawan
- Mengurangi trauma karena perubahan sistem. Klien dibiasakan perlahan-lahan menggunakan produknya bagian per bagian
- Memaksimalkan pengembalian modal investasi konsumen

## Kekurangan dari Incremental Model :

- Cocok untuk proyek berukuran kecil (tidak lebih dari 200.000 baris coding)
- Mungkin terjadi kesulitan untuk memetakan kebutuhan pengguna ke dalam rencana spesifikasi masing-masing hasil increment
- Dapat menjadi build and Fix Model, karena kemampuannya untuk selalu mendapat perubahan selama proses rekayasa berlangsung

# Model Spiral / Model Boehm



Model ini cukup baru ditemukan, yaitu tahun 1988 oleh Barry Boehm.

Model ini mengadaptasi dua model perangkat lunak yang ada yaitu model prototyping dengan pengulangannya dan model waterfall dengan pengendalian dan sistematikanya.

Pengembang dalam model ini memadupadankan beberapa model umum tersebut untuk menghasilkan produk khusus atau untuk menjawab persoalan-persoalan tertentu selama proses pengerjaan proyek.

# MODEL SPIRAL / MODEL BOEHM

## KELEBIHAN MODEL SPIRAL :

- DAPAT DISESUAIKAN AGAR PERANGKAT LUNAK BISA DIPAKAI SELAMA HIDUP PERANGKAT LUNAK KOMPUTER.
- LEBIH COCOK UNTUK PENGEMBANGAN SISTEM DAN PERANGKAT LUNAK SKALA BESAR
- PENGEMBANG DAN PEMAKAI DAPAT LEBIH MUDAH MEMAHAMI DAN BEREAKSI TERHADAP RESIKO SETIAP TINGKAT EVOLUSI KARENA PERANGKAT LUNAK TERUS BEKERJA SELAMA PROSES

## KELEMAHAN MODEL SPIRAL:

- SULIT UNTUK MENYAKINKAN PELANGGAN BAHWA PENDEKATAN EVOLUSIONER INI BISA DIKONTROL.
- MEMERLUKAN PENAKSIRAN RESIKO YANG MASUK AKAL DAN AKAN MENJADI MASALAH YANG SERIUS JIKA RESIKO MAYOR TIDAK DITEMUKAN DAN DIATUR.
- BUTUH WAKTU LAMA UNTUK MENERAPKAN PARADIGMA INI MENUJU KEPASTIAN YANG ABSOLUTE



# **PERBEDAAN ANTARA RPL DENGAN RS (REKAYASA SISTEM)**

- **REKAYASA SISTEM BERKAITAN DENGAN SEMUA ASPEK DALAM PEMBANGUNAN SISTEM BERBASIS KOMPUTER TERMASUK HARDWARE, REKAYASA PL DAN PROSES.**
- **RPL ADALAH BAGIAN DARI REKAYASA SISTEM YANG MELIPUTI PEMBANGUNAN PL, INFRASTRUKTUR, KONTROL, APLIKASI DAN DATABASE PADA SISTEM.**