METODOLOGI PENGEMBANGAN PERANGKAT LUNAK

Telkom University

Budhi Irawan, S.Si, M.T

PENDAHULUAN

- *Pengembangan perangkat lunak* dapat diartikan **sebagai proses** membuat suatu perangkat lunak baru untuk menggantikan perangkat lunak lama secara keseluruhan atau **memperbaiki** perangkat lunak yang telah ada.
- *Metodologi pengembangan perangkat lunak* adalah *suatu proses pengorganisasian kumpulan metode dan konvensi notasi yang telah didefinisikan untuk mengembangkan perangkat lunak*.

TUJUAN PENGEMBANGAN

- Secara prinsip bertujuan untuk *membantu menghasilkan perangkat lunak yang berkualitas*.
- *Penggunaan suatu metodologi sesuai dengan persoalan yang akan dipecahkan* dan *memenuhi kebutuhan pengguna akan menghasilkan suatu produk perekayasaan yang berkualitas dan terpelihara* serta dapat menghindari masalah-masalah yang sering terjadi seperti *estimasi penjadwalan dan biaya*, *perangkat lunak yang tidak sesuai dengan keinginan pengguna* dan sebagainya.

KOMPONEN PENGEMBANGAN

Menurut Pressman (1997) Komponen metodologi pengembangan perangkat lunak dapat dibagi dalam tiga unit, yaitu :

1. **Metode**, yaitu suatu cara atau teknik pendekatan yang sistematik yang dipergunakan untuk mengembangkan perangkat lunak. **Metode** ini mencakup : ***Perencanaan proyek dan perkiraan***, ***analisis keperluan sistem dan perangkat lunak***, ***perancangan struktur data***, ***arsitektur program***, ***prosedur algoritma***, ***Coding***, ***uji coba*** dan ***pemeliharaan***.
2. **Alat bantu (Tools)**, yaitu alat-alat (manual atau otomatis) yang mendukung pengembangan perangkat lunak. Terdapat 2 alat Bantu yang dapat digunakan yaitu : ***alat Bantu manual*** dan ***alat Bantu otomatis***.
3. **Prosedur**, yang dipergunakan untuk mendefinisikan urut-urutan pekerjaan (daur) dari metode dan alat bantu tersebut.

MODEL PROSES PENGEMBANGAN

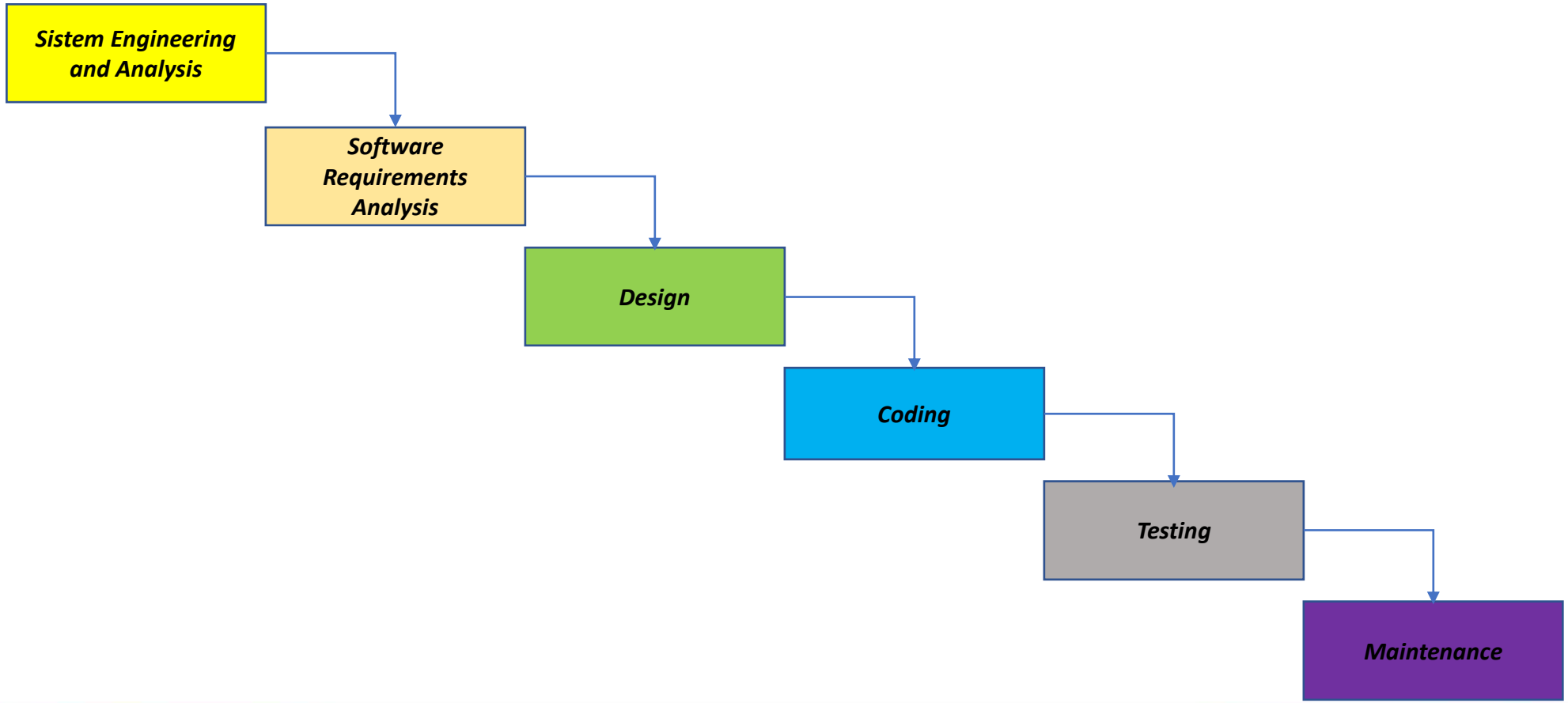
A. Linear Sequential Model
B. Prototyping Model
C. RAD (Rapid Application Development) Model
D. Spiral Model
E. Fourth Generation Techniques (4GT)

LINEAR SEQUENTIAL MODEL

Linear sequential model (atau disebut juga "***classic life cycle***" atau "***waterfall model***") adalah metode pengembangan perangkat lunak dengan pendekatan ***sekuensial*** dan merupakan model pengembangan perangkat lunak paling tua, dan paling banyak dipakai dengan cakupan aktivitas :

1. Rekayasa Sistem dan Analisis (*Sistem Engineering and Analysis*)
2. Analisis Kebutuhan Perangkat Lunak (*Software Requirements Analysis*)
3. Perancangan (*Design*)
4. Pembuatan Kode (*Coding*)
5. Pengujian (*Testing*)
6. Pemeliharaan (*Maintenance*)

LINEAR SEQUENTIAL MODEL



LINEAR SEQUENTIAL MODEL

Pemeliharaan *(Maintenance)* Meliputi :

1. ***Corrective Maintenance*** : Mengoreksi kesalahan pada perangkat lunak, yang baru terdeteksi pada saat perangkat lunak dipergunakan
2. ***Adaptive Maintenance*** : Penyesuaian dengan lingkungan baru, misalnya sistem operasi atau sebagai tuntutan atas perkembangan sistem komputer, misalnya penambahan *printer driver*
3. ***Perfektive Maintenance*** : Bila perangkat lunak sukses dipergunakan oleh pemakai. Pemeliharaan ditujukan untuk menambah kemampuannya seperti memberikan fungsi-fungsi tambahan, peningkatan kinerja dan sebagainya.

LINEAR SEQUENTIAL MODEL

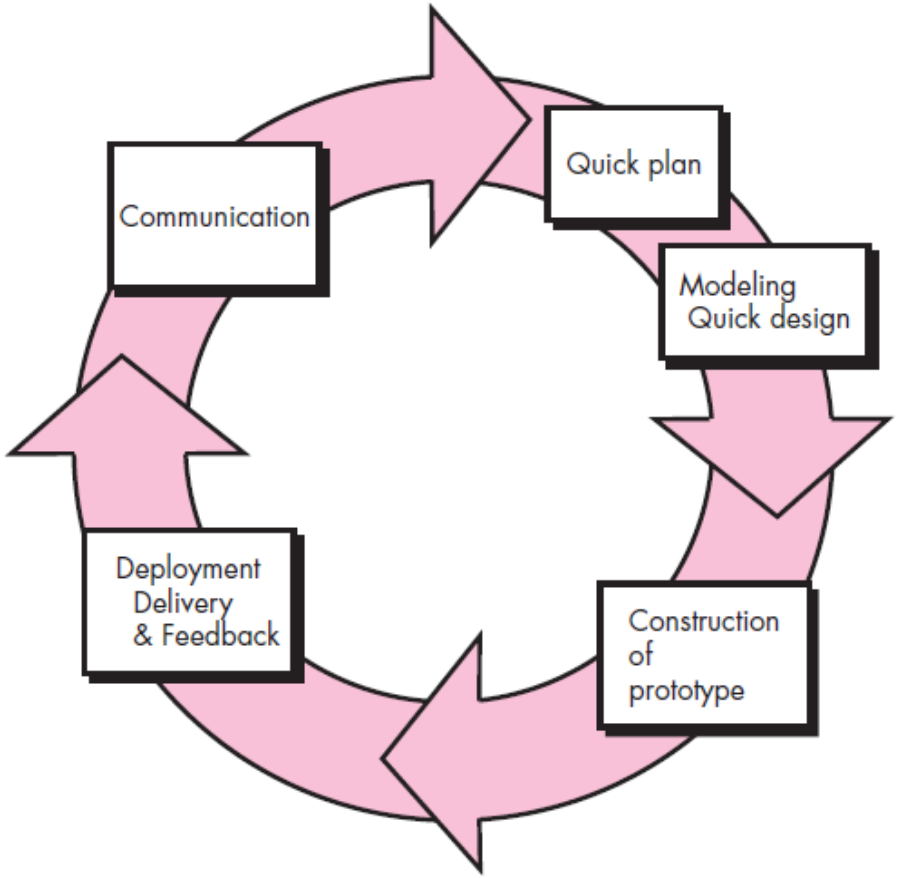
Kelemahan model *linear sequential*:

1. Proyek yang sebenarnya jarang mengikuti alur sekuensial seperti diusulkan, sehingga perubahan yang terjadi dapat menyebabkan hasil yang sudah didapat tim harus diubah kembali/iterasi sering menyebabkan masalah baru.
2. Linear sequential model mengharuskan semua kebutuhan pemakai sudah dinyatakan secara eksplisit di awal proses, tetapi kadang-kadang ini tidak dapat terlaksana karena kesulitan yang dialami pemakai saat akan mengungkapkan semua kebutuhannya tersebut.
3. Pemakai harus bersabar karena versi dari program tidak akan didapat sampai akhir rentang waktu proyek.
4. Adanya waktu menganggur bagi pengembang, karena harus menunggu anggota tim proyek lainnya menuntaskan pekerjaannya.

PROTOTYPING MODEL

Pendekatan ***prototyping model*** digunakan jika pemakai hanya mendefenisikan **objektif umum dari perangkat lunak** tanpa merinci kebutuhan ***input, pemrosesan*** dan ***outputnya***, sementara pengembang tidak begitu yakin akan ***efesiensi algoritma***, ***adaptasi sistem operasi***, atau ***bentuk antarmuka manusia-mesin*** yang harus diambil.

PROTOTYPING MODEL



PROTOTYPING MODEL

Cakupan aktivitas dari *prototyping* model terdiri dari :

1. ***Mendefinisikan objektif secara keseluruhan*** dan ***mengidentifikasi kebutuhan yang sudah diketahui***.
2. Melakukan ***perancangan secara cepat*** sebagai dasar untuk ***membuat prototype***.
3. ***Menguji coba*** dan ***mengevaluasi prototype*** dan ***kemudian melakukan penambahan*** dan ***perbaikan-perbaikan terhadap prototype*** yang sudah dibuat.

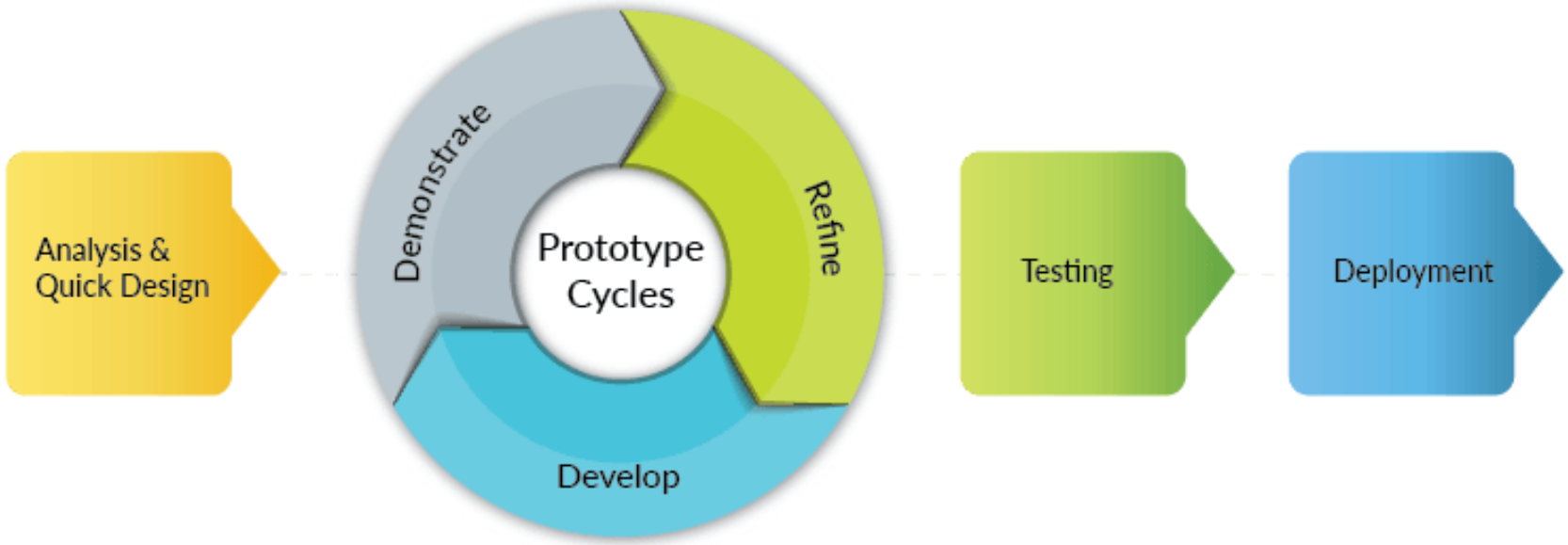
KELEMAHAN PROTOTYPING MODEL

1. Pelanggan yang ***melihat working version*** dari model yang dimintanya tidak menyadari, bahwa mungkin saja ***prototype dibuat terburu-buru*** dan ***rancangan tidak tersusun dengan baik***
2. Pengembang kadang-kadang ***membuat implementasi sembarang***, karena ***ingin working version bekerja dengan cepat***

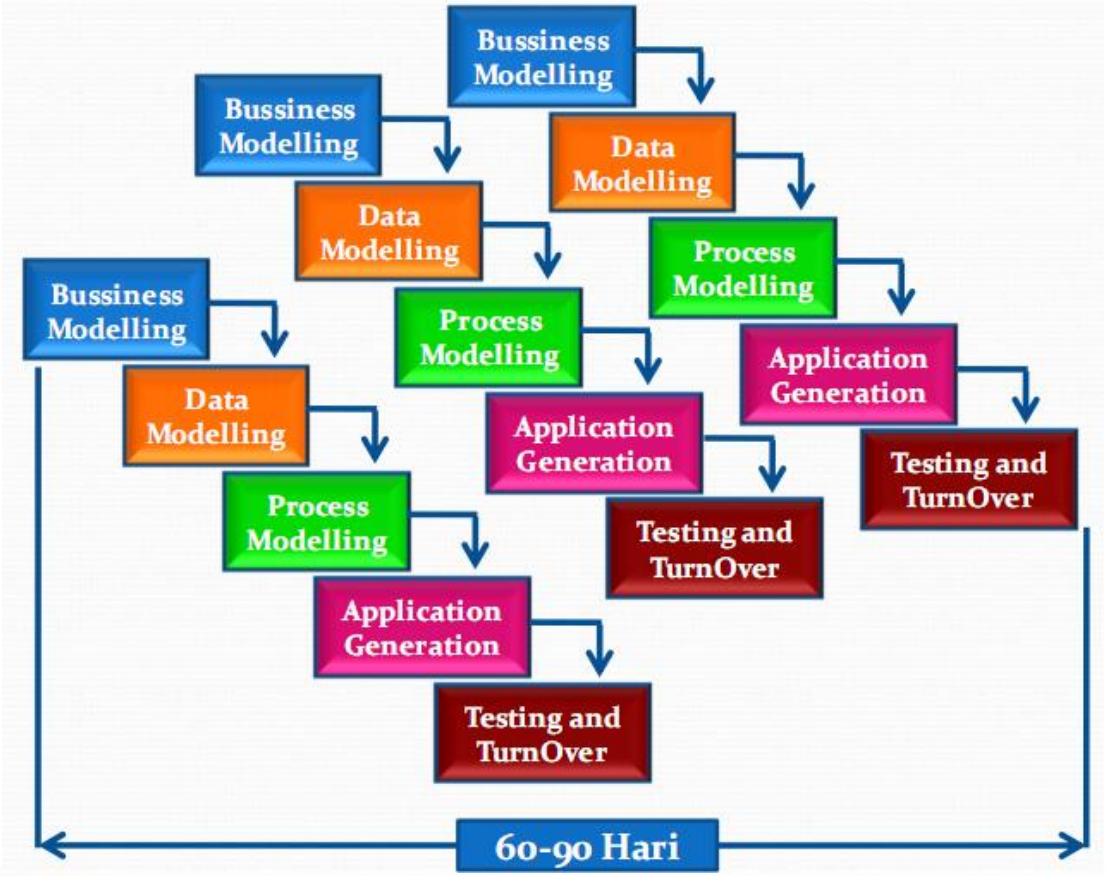
RAD (RAPID APPLICATION DEVELOPMENT) MODEL

Merupakan ***model proses pengembangan perangkat lunak secara linear sequential*** yang ***menekankan pada siklus pengembangan yang sangat singkat***. Jika kebutuhan dipahami dengan baik, proses RAD memungkinkan tim pengembangan menciptakan "***sistem fungsional yang utuh***" dalam periode waktu yang sangat pendek (kira-kira 60-90 hari).

RAD (RAPID APPLICATION DEVELOPMENT) MODEL



RAD (RAPID APPLICATION DEVELOPMENT) MODEL



RAD (RAPID APPLICATION DEVELOPMENT) MODEL

Pendekatan RAD model menekankan cakupan :

1. Pemodelan Bisnis (*Bussiness Modelling*)
2. Pemodelan Data (*Data Modelling*)
3. Pemodelan Proses (*Process Modelling*)
4. Pembuatan Aplikasi (*Application generation*)
5. Pengujian dan Pergantian (*Testing and turnover*)

RAD (RAPID APPLICATION DEVELOPMENT) MODEL

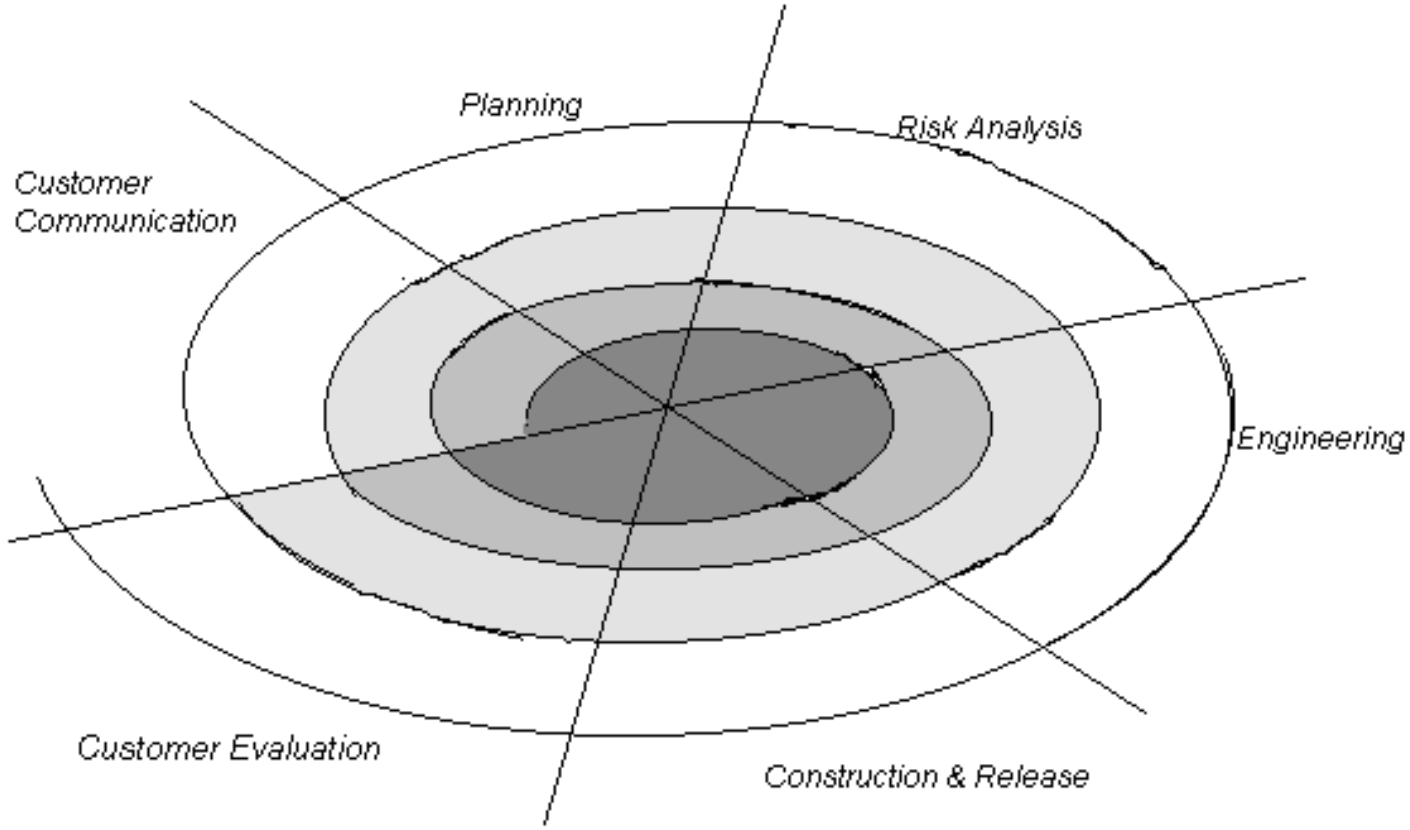
Kelemahan RAD model :

1. Untuk proyek dengan ***skala besar, RAD membutuhkan sumber daya manusia yang cukup untuk membentuk sejumlah tim RAD***.
2. RAD membutuhkan pengembang dan pemakai yang mempunyai ***komitmen untuk melaksanakan aktivitas melengkapi sistem*** dalam kerangka waktu yang singkat.
3. Akan menimbulkan masalah jika sistem ***tidak dapat dibuat secara modular***.
4. RAD tidak cocok digunakan untuk sistem yang ***mempunyai resiko teknik yang tinggi***.

SPIRAL MODEL

Merupakan ***model proses perangkat lunak yang memadukan wujud pengulangan*** dari ***model prototyping*** dengan ***aspek pengendalian dan sistematika dari linear sequential model***, dengan penambahan elemen baru yaitu ***analisis resiko***

SPIRAL MODEL



SPIRAL MODEL

Model ini memiliki 4 aktivitas penting, yaitu :

1. Perencanaan (*Planning*), penentuan tujuan, alternatif dan batasan
2. *Analisis resiko (Risk Analysis)*, analisis alternatif dan identifikasi/pemecahan resiko
3. *Rekayasa (Engineering)*, pengembangan level berikutnya dari produk
4. *Evaluasi Pemakai (Customer Evaluation)* penilaian terhadap hasil rekayasa

SPIRAL MODEL

- **Bentuk spiral** memberikan gambaran bahwa ***semakin besar iterasinya, maka menunjukkan makin lengkap versi dari perangkat lunak yang dibuat***.
- Selama awal sirkuit, objektif, alternatif dan batasan didefinisikan serta resiko diidentifikasikan dan dianalisa.
- ***Jika resiko menunjukkan ada ketidakpastian terhadap kebutuhan***, maka ***prototyping harus dibuat pada kuadran rekayasa***.
- Simulasi dan pemodelan lain dapat digunakan untuk mendefinisikan masalah dan memperbaiki kebutuhan.

SPIRAL MODEL

Kelemahan spiral model :

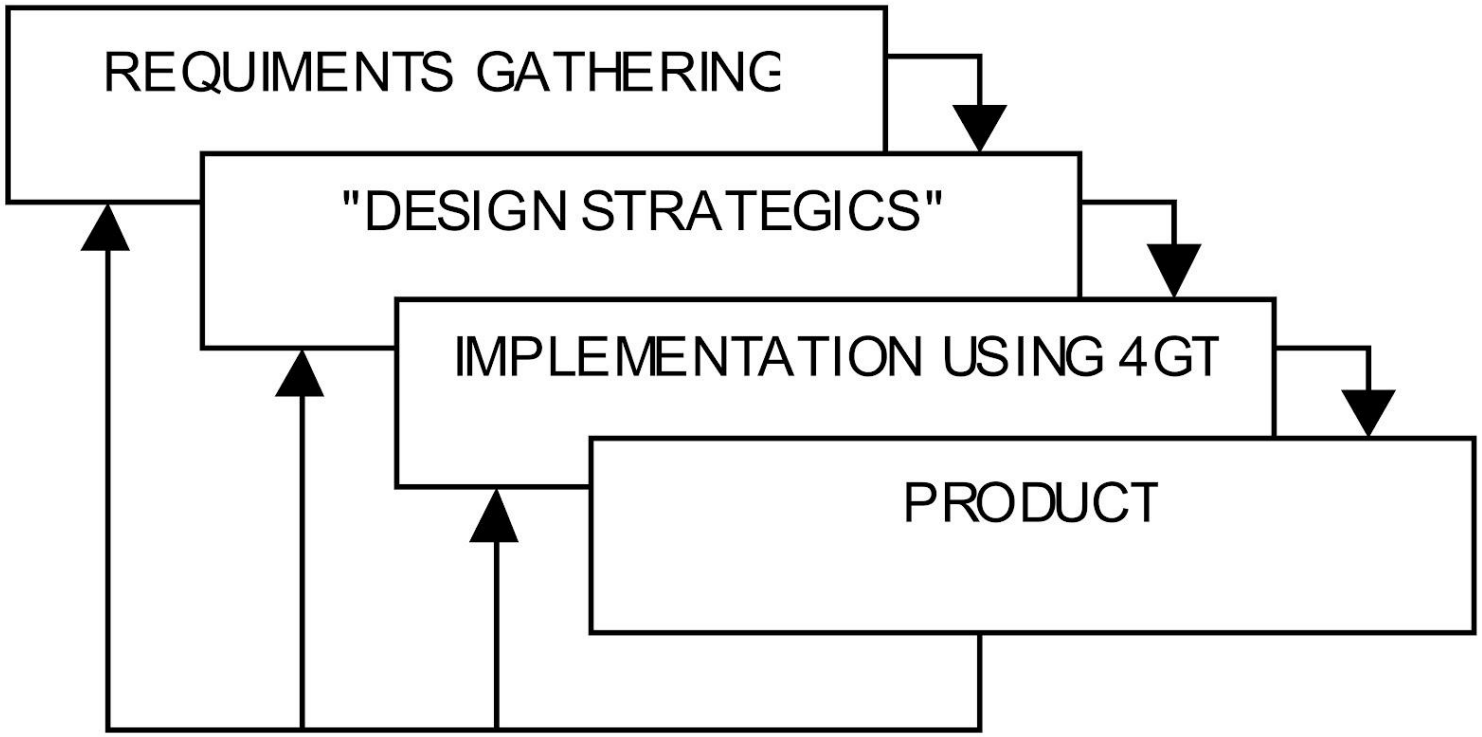
1. ***Sulit untuk meyakinkan pemakai*** (saat situasi kontrak) bahwa penggunaan pendekatan ini akan dapat dikendalikan.
2. ***Memerlukan tenaga ahli untuk memperkirakan resiko***, dan harus mengandalkannya supaya sukses.
3. ***Belum terbukti apakah metode ini cukup efisien*** karena usianya relatif baru.

FOURTH GENERATION TECHNIQUES (4GT)

Istilah generasi ke empat, mengarah ke perangkat lunak yang umum yaitu tiap pengembang perangkat lunak menentukan beberapa ***karakteristik perangkat lunak pada level tinggi***. Saat ini pengembangan perangkat lunak yang mendukung 4GT, berisi tool-tool berikut :

- Bahasa non prosedural untuk query basis data
- *Report generation*
- *Data manipulation*
- Interaksi layar
- Kemampuan grafik level tinggi
- Kemampuan spreadsheet

FOURTH GENERATION TECHNIQUES (4GT)



FOURTH GENERATION TECHNIQUES (4GT)

Cakupan aktivitas 4GT :

1. Pengumpulan kebutuhan, idealnya pelanggan akan menjelaskan kebutuhan yang akan ditranslasikan ke prototype operasional.
2. Translasi kebutuhan menjadi prototype operasional, atau langsung melakukan implementasi secara langsung dengan menggunakan bahasa generasi keempat (4GL) jika aplikasi relatif kecil.
3. Untuk aplikasi yang cukup besar, dibutuhkan strategi perancangan sistem walaupun 4GL akan digunakan.
4. Pengujian.
5. Membuat dokumentasi.
6. Melaksanakan seluruh aktivitas untuk mengintegrasikan solusi-solusi yang membutuhkan paradigma rekayasa perangkat lunak lainnya.

TUGAS KELOMPOK

Lakukan analisis terhadap ke 5 model pengembangan perangkat lunak :

A. Linear Sequential Model
B. Prototyping Model
C. RAD (Rapid Application Development) Model
D. Spiral Model
E. Fourth Generation Techniques (4GT)

Kapan kita menggunakan salah satu model tersebut ? Mana yang paling banyak digunakan ? Berikan contoh masing-masing penerapannya ? dan bandingkan kekurangan dan kelemahannya diantara model tersebut?