



## **Pertemuan 2:**

# **Process Model**

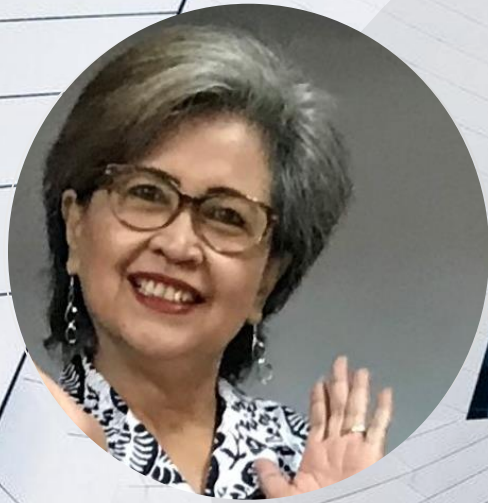
**Meuthia Rachmaniah**

**Departemen Ilmu Komputer, FMIPA IPB**

*Software Engineering: A Practitioner's Approach, Ninth Ed.*

*Roger S. Pressman dan Bruce R. Maxim*

Copyright © 2020 McGraw-Hill Education



# Hello!

Ir. Meuthia Rachmaniah, M.Sc.  
*Associate Professor*  
NIP 195907111984032006

Computer Science Department  
Software Engineering & Information Sciences  
[meuthiara@apps.ipb.ac.id](mailto:meuthiara@apps.ipb.ac.id)

SATYALANCANA KARYA SATYA 20 TAHUN  
SATYALANCANA KARYA SATYA 30 TAHUN

# Cakupan Materi



- 2.1 Model Proses Generik
- 2.2 Mendefinisikan *Framework Activity*
- 2.3 Mengidentifikasi Gugus Tugas (*Task Set*)
- 2.4 Penilaian dan Peningkatan Proses (*Process Assessment and Improvement*)
- 2.5 Model Proses Preskriptif
- 2.6 Product and Process
- 2.7 Rangkuman

# Quick Look

## WHAT IS IT?



- Saat membangun produk atau sistem, penting untuk mengikuti serangkaian langkah yang dapat diprediksi (peta jalan) yang membantu Anda mengirimkan produk berkualitas tinggi tepat waktu
- Peta jalan ini disebut “Software Process.”

## WHO DOES IT



- Perekraya PL menyesuaikan proses dengan kebutuhan klien dan kemudian mengikutinya
- Orang yang meminta PL juga memiliki peran dalam proses mendefinisikan, membangun, dan mengujinya

## HOW DO I ENSURE THAT I'VE DONE IT RIGHT



- Kualitas, ketepatan waktu, dan kelangsungan hidup jangka panjang dari produk yang dibangun adalah indikator terbaik dari keberhasilan proses yang digunakan



## WHAT ARE THE STEPS

- Proses yang Anda adopsi bergantung pada perangkat lunak yang Anda buat.
- Suatu proses mungkin sesuai untuk membuat perangkat lunak untuk sistem avionik pesawat tetapi mungkin tidak berfungsi dengan baik untuk pembuatan aplikasi seluler atau video game



## WHY IS IT IMPORTANT?

- Sebuah proses memberikan stabilitas, kontrol, dan organisasi pada suatu aktivitas sehingga tidak menjadi kacau.
- Proses RPL modern harus AGILE
- Harus mencakup aktivitas, kontrol, dan work product yang untuk tim proyek dan produk yang akan diproduksi



## WHAT IS THE WORK PRODUCT

- Work product ialah program, dokumen, dan data yang dihasilkan dari aktivitas dan task rekayasa yang dicakup dalam proses

# Social Learning Process (Proses Pembelajaran Sosial)





# Definisi Proses Perangkat Lunak

## Proses Perangkat Lunak

Proses PL mendefinisikan pendekatan yang diambil saat PL direkayasa



## Framework

**Framework** adalah **aktivitas**, **aksi** (action), dan **tugas** (task) yang diperlukan untuk membangun PL berkualitas tinggi

## Proses PL vs RPL

Proses PL tidak sama dengan **RPL**, yang juga mencakup teknologi yang mengisi proses— **metode teknis** dan **tool otomatis** (*automated tools*)



## 2.1 Model Proses Generik

# Model Proses Generik

- Framework Proses generik untuk RPL mendefinisikan 5 aktivitas framework:
  - Komunikasi, Perencanaan, Pemodelan, Konstruksi, dan Penerapan (Deployment)
- Gugus Umbrella Activities yang terdiri dari:
  - *Project Tracking and Control*, Manajemen Risiko, *Quality Assurance*, Manajemen Konfigurasi, Review Teknis, dan lainnya yang diterapkan di seluruh proses
- Alur Proses (Process Flow):
  - Adalah bagaimana Framework Activity dan actions dan tasks yang terdapat dalam setiap aktivitas dikelola sesuai urutan dan waktunya

## Software process

### Process framework

#### Umbrella activities

##### Framework activity #1

software engineering action #1.1

Task sets

work tasks  
work products  
quality assurance points  
project milestones

:

Software engineering action #1.k

Task sets

work tasks  
work products  
quality assurance points  
project milestones

:

##### Framework activity #n

software engineering action #n.1

Task sets

work tasks  
work products  
quality assurance points  
project milestones

:

Software engineering action #n.m

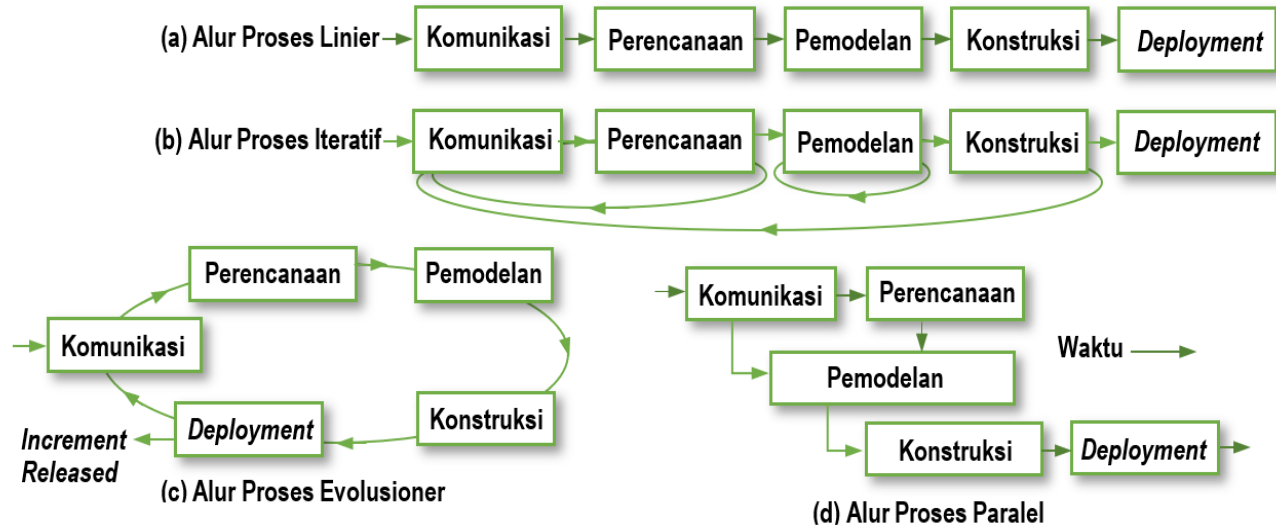
Task sets

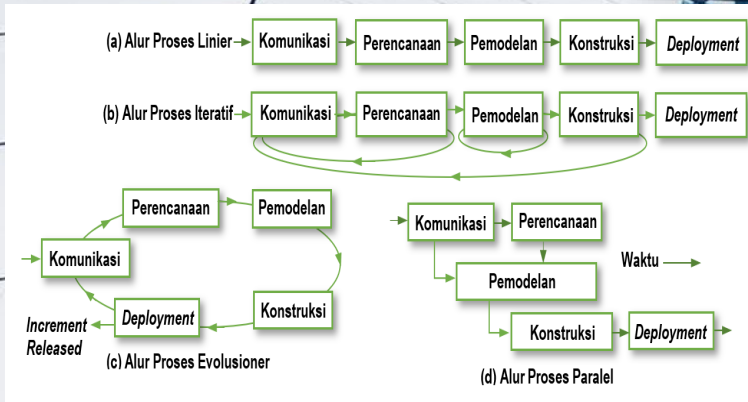
work tasks  
work products  
quality assurance points  
project milestones



# Alur Proses (*Process Flow*)

Alur Proses (Process Flow) menjelaskan bagaimana Framework Activity dan Action serta Task yang terjadi dalam setiap Framework Activity diatur sesuai dengan urutan dan waktu





- Alur Proses Linear
  - Mengeksekusi setiap aktivitas dari lima aktivitas secara berurutan
- Alur Proses Iteratif
  - Mengulang satu atau lebih aktivitas sebelum berlanjut ke aktivitas berikutnya
- Alur Proses Evolusioner
  - Mengeksekusi aktivitas secara sirkular
  - Setiap sirkular menuju ke versi perangkat lunak yang lebih lengkap
- Alur Proses Paralel
  - Mengeksekusi satu atau lebih aktivitas secara paralel dengan aktivitas lainnya
  - Misal: aspek dari pemodelan perangkat lunak dilakukan secara paralel dengan aspek lain dari perangkat lunak)



## 2.2 Mendefinisikan Framework Activity

Tim PL akan membutuhkan **lebih banyak informasi** secara signifikan sebelum dapat menjalankan salah satu dari aktivitas dari proses PL dengan benar



○ Harus dapat **menjawab pertanyaan**:

- Action apa yang sesuai untuk Framework Activity, mengingat
  - **Sifat problem** yang harus dipecahkan,
  - **Karakteristik** orang yang melakukan pekerjaan, dan
  - **Stakeholder** yang mensponsori proyek?

## Small Project

- Untuk proyek PL kecil yang diminta oleh satu orang (di lokasi remote) dengan kebutuhan sederhana dan langsung,
  - Aktivitas komunikasi mungkin mencakup sedikit lebih dari panggilan telepon atau email dengan stakeholder yang meminta proyek



**Aktivitas:** *Komunikasi*

**Action:** *Percakapan telepon*

**Tasks:**

1. Kontak stakeholder dengan telepon
2. Diskusikan kebutuhan /persyaratan (*requirements*) dan buat catatan
3. Kelola catatan menjadi pernyataan tertulis singkat tentang kebutuhan
4. Email stakeholder agar direviu dan disetujui

## More Complex Project:

- Stakeholder banyak
- Setiap stakeholder berbeda kebutuhan/*requirements* (terkadang berbenturan), maka aktivitas Komunikasi terdiri dari 6 (enam) actions.
- *Masing-masing Action mungkin memiliki banyak Task dan dalam beberapa kasus sejumlah Work Product yang berbeda*



**Aktivitas:** *Komunikasi*

**Action:**

1. Inception
2. Elicitation
3. Elaboration
4. Negotiation
5. Specification
6. Validation

**Tasks:**

- 1.1 ...1.n (untuk Action 1)
- 1.2 ...2.n (untuk Action 2)
- ...



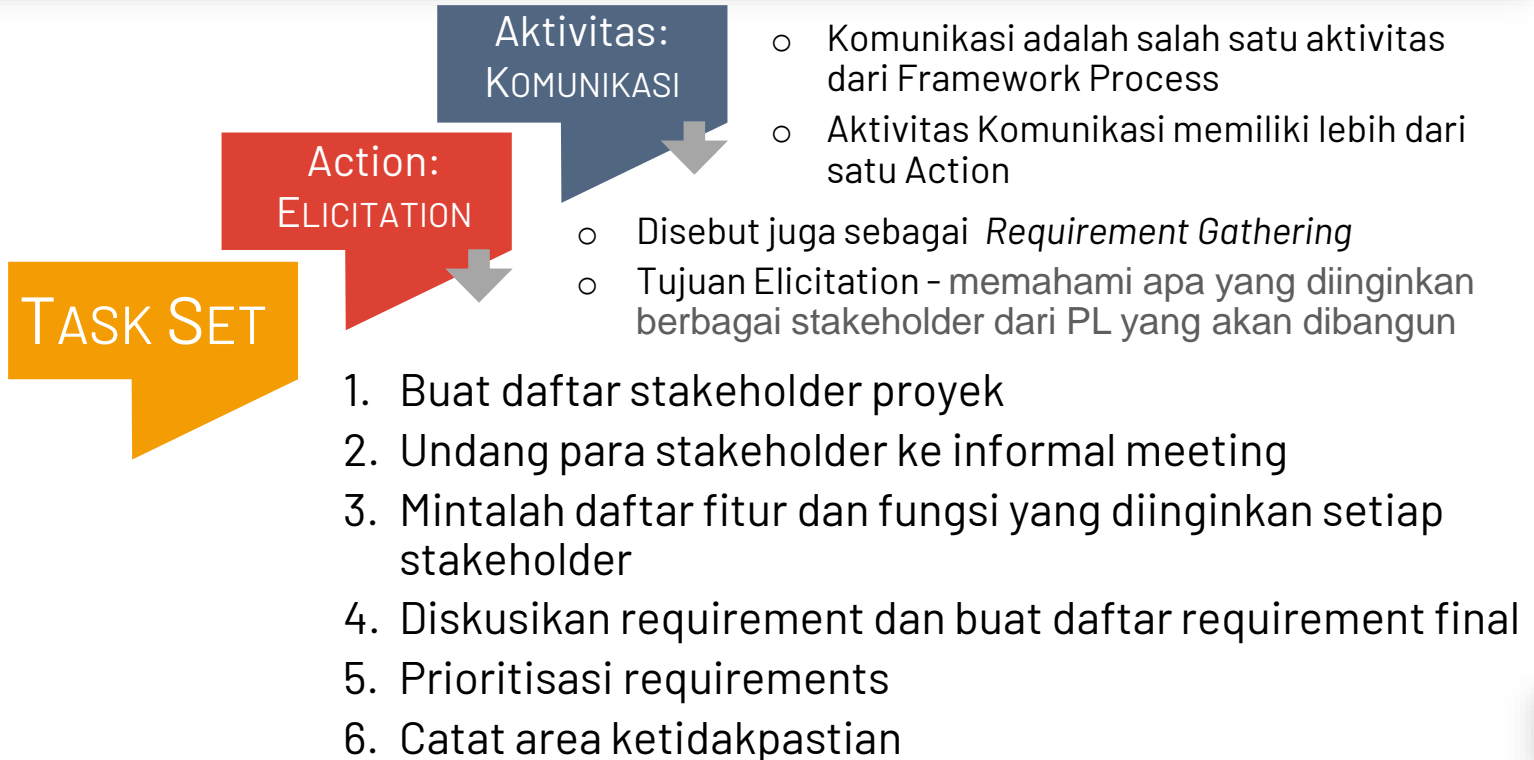


## 2.3 Mengidentifikasi Gugus Tugas (Tasks Set)



- Satu gugus task (tugas) mendefinisikan pekerjaan aktual yang harus dilakukan untuk mencapai tujuan dari Action
  - Daftar Task untuk dicapai
  - Daftar Work Product untuk dihasilkan
  - Daftar penjaminan mutu (*quality assurance*) yang harus diterapkan
  - *Project Milestone* yang dihasilkan
- Proyek berbeda akan meminta task set yang berbeda
  - Artinya, Action harus disesuaikan dengan kebutuhan spesifik proyek PL dan karakteristik tim proyek

# Task Set untuk **Simple Project** Banyak Stakeholder



# Task Set untuk **BIG Project**

## Aktivitas: KOMUNIKASI

- Komunikasi adalah salah satu aktivitas dari Framework Process
- Aktivitas Komunikasi memiliki lebih dari satu Action

## Action: ELICITATION

- Disebut juga sebagai *Requirement Gathering*
- Tujuan Elicitation - memahami apa yang diinginkan berbagai stakeholder dari PL yang akan dibangun

## TASK SET

1. Buat daftar stakeholder proyek
2. Wawancarai setiap stakeholder secara terpisah untuk menentukan keseluruhan keinginan dan kebutuhan
3. Buat daftar awal fungsi dan fitur berdasarkan masukan stakeholder
4. Jadwalkan serangkaian pertemuan spesifikasi aplikasi yang difasilitasi
5. Melakukan pertemuan (*meeting*)
6. Hasilkan skenario pengguna informal sebagai bagian dari setiap pertemuan
7. Perbaiki skenario pengguna berdasarkan umpan balik stakeholder
8. Buat daftar revisi requirement stakeholder
9. Gunakan teknik penyebaran fungsi kualitas untuk memprioritaskan kebutuhan
10. *Package requirement* agar dapat dikirimkan secara bertahap
11. Perhatikan kendala dan batasan yang akan ditempatkan pada sistem
12. Diskusikan metode untuk memvalidasi sistem

○ Task Set slide #16 dan #17 memenuhi "pengumpulan kebutuhan" (*requirement gathering*), tetapi sangat berbeda kedalaman dan tingkat formalitasnya.

○ Tim PL memilih Task Set yang memungkinkan untuk mencapai tujuan untuk setiap Action dan tetap mempertahankan KUALITAS dan tetap AGILE



## 2.4 Penilaian dan Peningkatan Proses

*(Process Assessment and Improvement)*

# Assesment dan Process Pattern

- Pola proses (process pattern) harus digabungkan dengan praktik RPL yang solid
- Proses dan aktivitas PL harus dinilai dengan menggunakan ukuran numerik atau analitik PL (metrik)
  - proses dapat dinilai untuk memastikan telah memenuhi serangkaian kriteria proses dasar yang telah terbukti penting untuk RPL yang sukses
- Pola Proses (Process Pattern)
  - Menggambarkan masalah terkait proses yang dihadapi selama pekerjaan RPL
  - Mengidentifikasi lingkungan di mana masalah telah dihadapi, dan
  - Menyarankan satu atau lebih solusi yang telah terbukti mengatasi masalah tersebut



# Type-tipe Process Pattern

## STAGE PATTERN

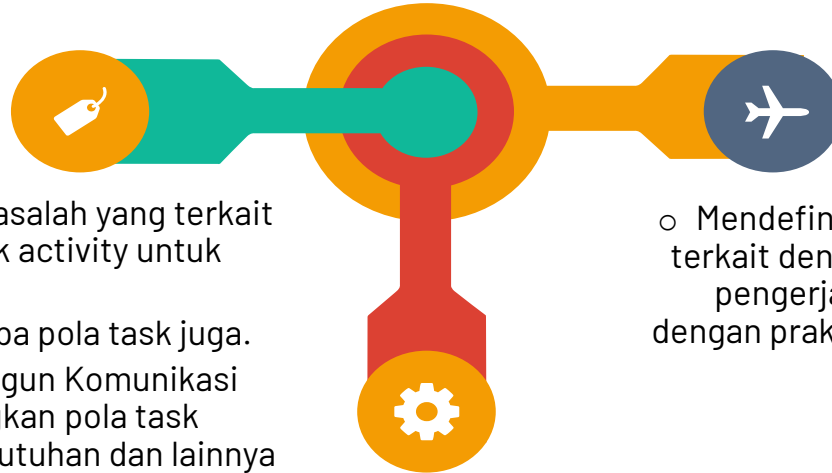
- Mendefinisikan masalah yang terkait dengan framework activity untuk proses tersebut.
- Mencakup beberapa pola task juga.
- Misalnya, Membangun Komunikasi akan menggabungkan pola task Pengumpulan kebutuhan dan lainnya

## TASK PATTERN

- Mendefinisikan masalah yang terkait dengan action RPL atau pengerjaan task dan relevan dengan praktik RPL yang sukses

## PHASE PATTERN

- Mendefinisikan urutan framework activity yang terjadi dengan proses, bahkan ketika keseluruhan aliran aktivitas bersifat iteratif
- Contohnya termasuk Spiral Model atau Prototyping



# Contoh Process Pattern

- Menjelaskan pendekatan yang dapat diterapkan ketika stakeholder memiliki gagasan umum tentang apa yang harus dilakukan tetapi tidak yakin dengan kebutuhan PL tertentu
- **PATTERN NAME.** kebutuhan (Requirement) Tidak Jelas
- **INTENT.** Pola ini menggambarkan pendekatan untuk membangun model yang dapat dinilai secara iteratif oleh stakeholder dalam upaya mengidentifikasi atau memantapkan kebutuhan PL
- **TYPE.** Phase pattern
- **INITIAL CONTEXT.** Kondisi yang harus dipenuhi (1) Stakeholder telah diidentifikasi; (2) Moda komunikasi antara stakeholder dan tim PL telah ditetapkan; (3) Masalah PL utama yang harus dipecahkan telah diidentifikasi oleh para stakeholder; (4) Pemahaman awal tentang ruang lingkup proyek, kebutuhan bisnis dasar, dan kendala proyek telah dikembangkan
- **PROBLEM.** Requirements samar-samar atau tidak ada. pemangku kepentingan tidak yakin dengan apa yang mereka inginkan.
- **SOLUTION.** Deskripsi proses pembuatan prototipe akan disajikan di sini.
- **RESULTING CONTEXT.** Prototipe PL yang mengidentifikasi kebutuhan dasar. (moda interaksi, fitur komputasi, fungsi pemrosesan) disetujui oleh stakeholder. Setelah ini, 1. Prototipe ini dapat berkembang melalui serangkaian peningkatan menjadi PL produksi atau 2. Prototipe dapat dibuang.
- **RELATED PATTERNS.** CustomerCommunication, IterativeDesign, IterativeDevelopment, CustomerAssessment, RequirementExtraction.

# Penilaian dan Peningkatan Proses

(Process Assessment and Improvement)

01

## SCAMPI

- Standard CMMI Assessment Method for Process Improvement (SCAMPI)
- Menyediakan model penilaian proses lima langkah yang menggabungkan lima fase: memulai, mendiagnosis, menetapkan, bertindak, dan belajar

03

## CBA IPI

- CMM-Based Appraisal for Internal Process Improvement (CBA IPI)
- Menyediakan teknik diagnostik untuk menilai kematangan relatif dari organisasi PL;
- Menggunakan SEI CMM sebagai dasar penilaian

02

## SPICE

- SPICE—The SPICE (ISO/IEC15504) Standard
- Mendefinisikan satu set kebutuhan untuk penilaian proses PL.
- Maksud dari standar ini adalah untuk membantu organisasi dalam mengembangkan evaluasi objektif dari kemandirian setiap proses PL yang ditentukan

04

## ISO 9001:2000

- ISO 9001:2000 for Software
- Standar generik yang berlaku untuk organisasi mana pun yang ingin meningkatkan kualitas keseluruhan produk, sistem, atau layanan yang disediakan.
- Standar ini secara langsung dapat diterapkan pada organisasi dan perusahaan PL



## 2.5 Model Proses Preskriptif

- 2.5.1 Model Waterfall
- 2.5.2 Model Incremental
- 2.5.3 Model Proses Prototyping
- 2.5.4 Model Proses Evolusioner
- 2.5.5 Model Concurrent
- 2.5.5 Model Proses Unified

# Model Proses Preskriptif



Model proses preskriptif mendefinisikan satu set elemen proses yang telah ditentukan sebelumnya dan alur kerja proses yang dapat diprediksi



Model proses preskriptif meminta untuk struktur dan ketertiban dalam pengembangan PL



Kegiatan dan task terjadi secara berurutan dengan pedoman yang ditetapkan untuk kemajuan



Setiap model proses juga menentukan alur proses (juga disebut alur kerja) – yaitu, cara elemen proses saling terkait satu sama lain



Disebut "preskriptif" karena meresepkan satu set elemen proses – aktivitas framework, action RPL, task, produk kerja, jaminan kualitas, dan mekanisme kontrol perubahan untuk setiap proyek

# Model Proses Preskriptif

## 2.5.1 Model Waterfall

- a. Model Waterfall
- b. Model V

## 2.5.2 Model Incremental

## 2.5.3 Model Proses Prototyping

## 2.5.4 Model Proses Evolusioner

- a. Prototyping
- b. Spiral Model

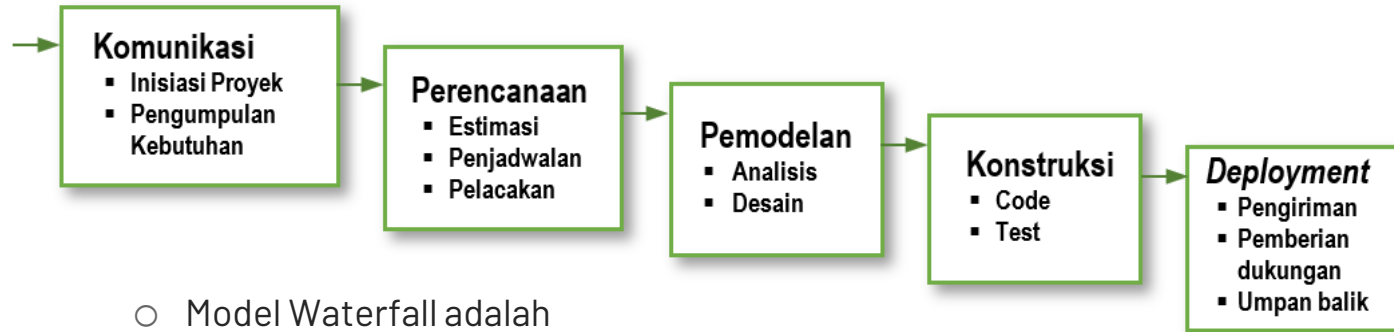
## 2.5.5 Model Concurrent

## 2.5.6 Model Proses Unified





## 2.5.1.a Model Waterfall

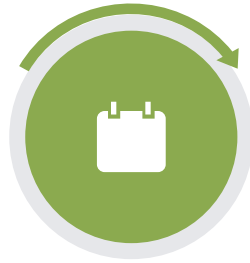


- Model Waterfall adalah **paradigma tertua** pada rekayasa perangkat lunak
- Model waterfall dapat berfungsi sebagai model proses pada situasi di mana **kebutuhan telah jelas** ditetapkan sehingga pekerjaan dapat dilanjutkan hingga tahap penyelesaian secara linier
- Model Waterfall dipilih apabila kebutuhan-kebutuhan **terdefinisi baik** dan **cukup stabil**, dan mengarah ke tipe **alur proses linier**, yaitu dimulai dengan komunikasi, perencanaan, pemodelan, konstruksi, dan diakhiri dengan *deployment*

# Permasalahan pada Model Waterfall

## Klien Sulit Menyatakan Requirements

Klien seringkali **sulit menyatakan semua kebutuhan** secara eksplisit. Padahal model waterfall membutuhkan kepastian pada saat awal proyek pengembangan sistem dimulai



## Jarang Linier dan Sulit Iterasi

Dalam realitanya, pengembangan sistem jarang mengikuti aliran sekuensial yang diusulkan model. Meskipun model linier dapat mengakomodasi iterasi, namun ini dilakukan secara tidak langsung, sehingga perubahan tak langsung ini dapat menyebabkan kebingungan saat tim proyek melanjutkan

## Blocking pada Aktivitas Sebelumnya

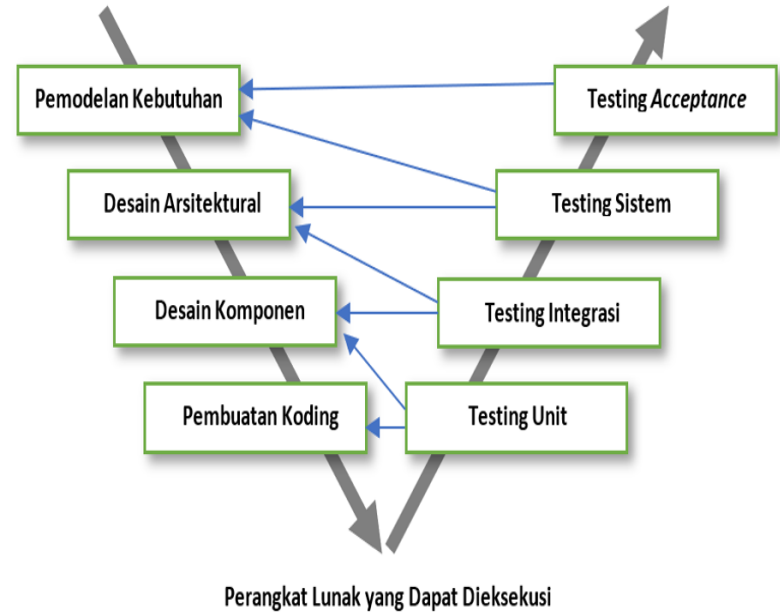
Terjadi *blocking state*, yaitu beberapa anggota tim proyek harus menunggu anggota tim lainnya untuk menyelesaikan tasks. Padahal, waktu yang dihabiskan untuk menunggu bisa melebihi waktu yang dihabiskan untuk pekerjaan produktif. *Blocking state* cenderung lebih umum terjadi di awal dan akhir proses sekuensial linier

## Versi kerja di akhir Proyek

Klien harus bersabar. Versi kerja dari program tidak akan tersedia hingga akhir jangka waktu proyek. Apabila terdapat kesalahan besar yang tidak terdeteksi maka hal ini dapat menjadi bencana

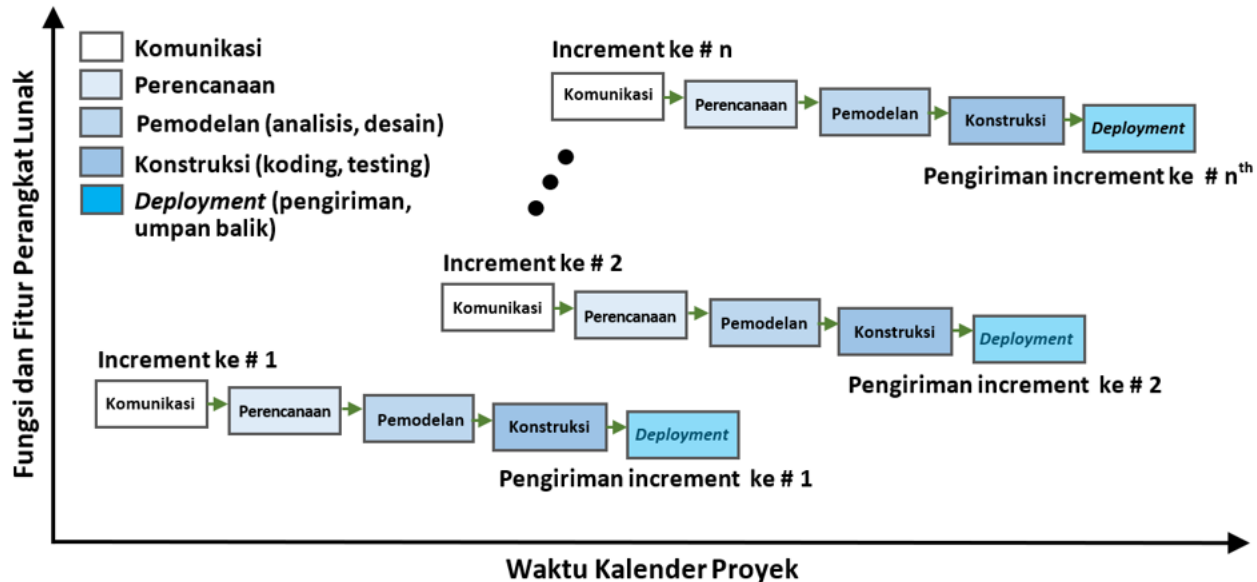
## 2.5.1.b Model V

- Variasi model Waterfall yang menggambarkan hubungan action jaminan kualitas dengan action yang terkait dengan komunikasi, pemodelan, dan pengaktifan konstruksi kode awal
- Tim pertama-tama bergerak ke sisi kiri V untuk memperbaiki kebutuhan masalah.
- Setelah kode dibuat, tim bergerak ke sisi kanan V, melakukan serangkaian tes yang memvalidasi setiap model yang dibuat saat tim bergerak ke sisi kiri.
- Pada kenyataannya, tidak ada perbedaan mendasar antara siklus hidup klasik dan model V.
- Dalam hal ini, model-V menyediakan cara untuk memvisualisasikan bagaimana verifikasi dan action validasi diterapkan pada awal pengembangan sistem

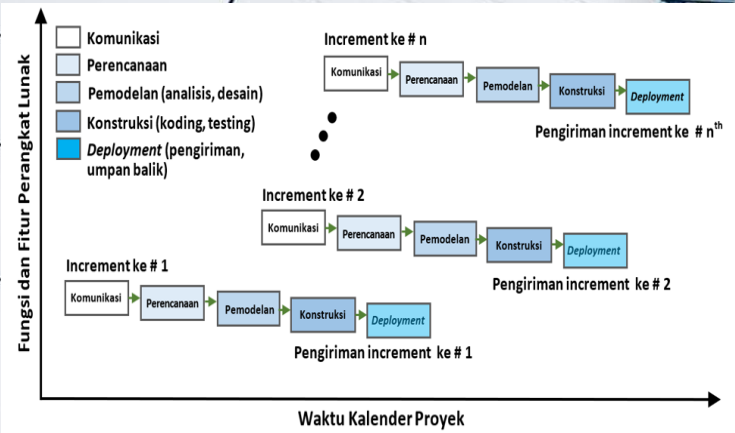


## 2.5.2 Model Incremental

- Model Proses *Incremental* dipilih ketika kebutuhan awal didefinisikan dengan cukup baik, tetapi keseluruhan ruang lingkup upaya pengembangan menghalangi proses yang murni linier
- Terdapat kebutuhan yang mendesak untuk memperluas rangkaian fungsi baru yang terbatas ke rilis sistem yang lebih baru.



## 2.5.2 Model Incremental



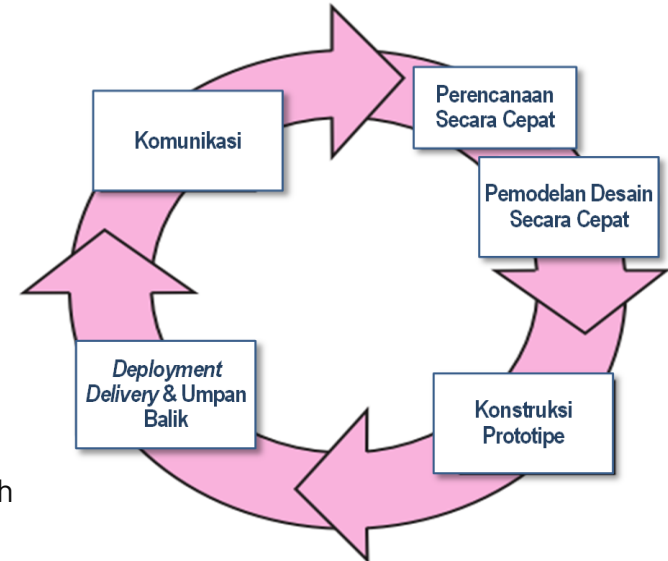
- Model Proses *Incremental* menggabungkan elemen aliran proses linier dan paralel
- Model *Incremental* menerapkan sekuens linear dalam *staggered fashion* sesuai waktu kalender berlangsung.
- Setiap urutan linier menghasilkan *deliverable increment* perangkat lunak dan dapat disampaikan kepada klien.
- *Increment* pertama sering merupakan produk inti dengan banyak fitur tambahan.
- Klien menggunakan dan mengevaluasi *increment* pertama.
- Klien juga memberikan umpan balik dengan lebih banyak modifikasi untuk lebih memenuhi kebutuhan.
- Aktivitas yang sama berlangsung sampai dengan *increment* ke- $n$

## 2.5.3 Prototyping Paradigm

### ☐ KAPAN DIGUNAKAN:

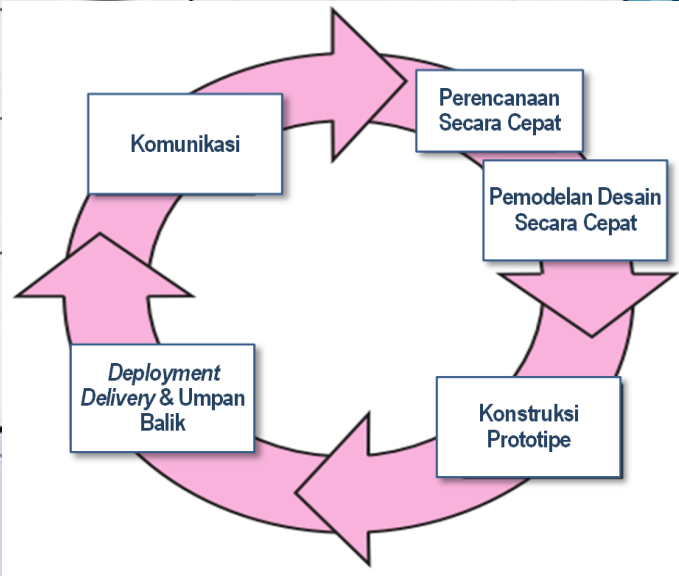
- Pelanggan menetapkan serangkaian tujuan umum tetapi tidak mengidentifikasi kebutuhan terperinci untuk fungsi dan fitur, atau
- Pengembang mungkin tidak yakin dengan efisiensi suatu algoritme, bentuk yang harus diambil oleh interaksi komputer manusia

- ☐ Baik pemangku kepentingan dan insinyur perangkat lunak menyukai paradigma prototyping.
- ☐ Pengguna dapat merasakan sistem yang sebenarnya, dan pengembang dapat segera membangun sesuatu.
- ☐ Namun, para insinyur dapat membuat kompromi untuk membuat prototipe bekerja dengan cepat.
- ☐ Pilihan yang kurang ideal dapat diadopsi selamanya setelah Anda terbiasa
- ☐ Iterasi akan terjadi sampai berbagai kebutuhan stakeholder dipenuhi. Pada saat bersamaan, tim pengembang juga akan lebih memahami apa yang perlu dilakukan





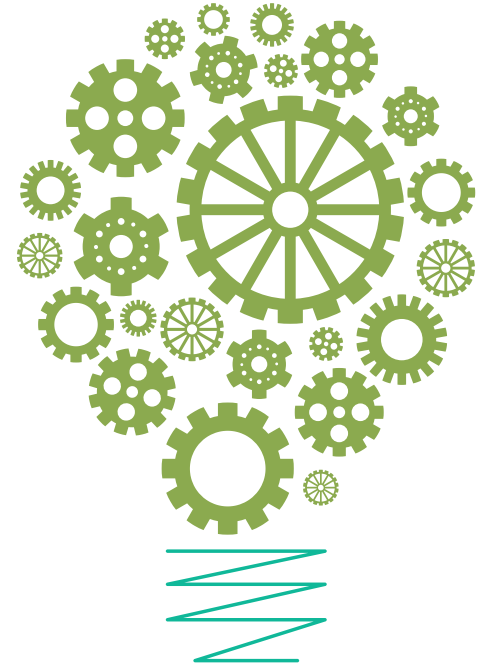
## Tahapan Prototyping Paradigm



- ☐ **KOMUNIKASI.** Meeting dengan stakeholder untuk menentukan tujuan, mengidentifikasi kebutuhan apa pun yang diketahui, garis besar area di mana definisi lebih lanjut adalah wajib
- ☐ **PERENCANAAN CEPAT.** Rencana cepat untuk pembuatan prototipe dan pemodelan (desain cepat) terjadi
- ☐ **PEMODELAN DESAIN CEPAT.** Berfokus pada representasi dari aspek-aspek perangkat lunak yang akan terlihat oleh pengguna akhir. (antarmuka/interface dan keluaran)
- ☐ **KONSTRUKSI PROTOTIPE.** Konstruksi prototipe yang akan digunakan dan dievaluasi
- ☐ **DEPLOYMENT DELIVERY & UMPAN BALIK.** Komentar pemangku kepentingan akan digunakan untuk menyempurnakan kebutuhan (requirements)

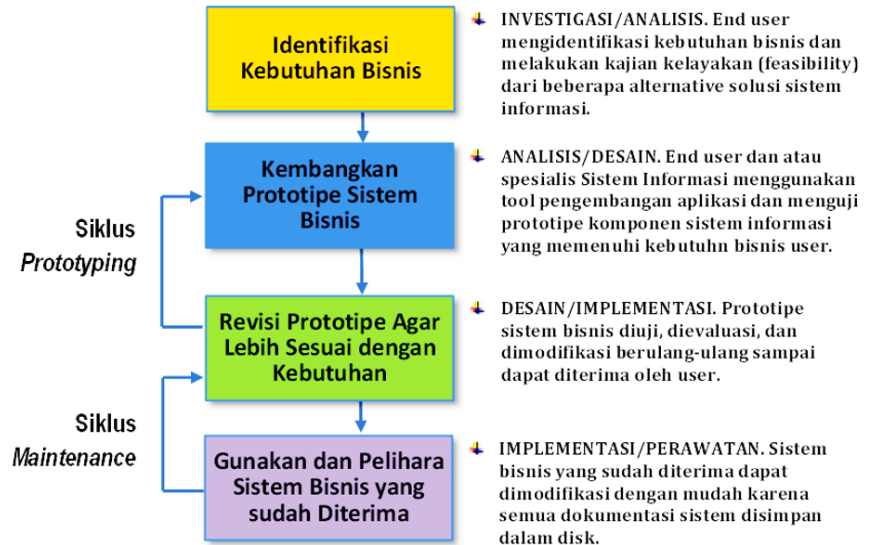
## 2.5.4. Model Proses Evolusioner

- Sistem PL berkembang dari waktu ke waktu akibat kebutuhan sering berubah sebagai hasil pengembangan. Dengan demikian, garis lurus ke produk akhir yang lengkap tidak mungkin. Namun, versi terbatas harus dikirimkan untuk memenuhi tekanan persaingan
- Biasanya satu set produk inti atau kebutuhan sistem dipahami dengan baik, tetapi detail dan ekstensinya belum ditentukan
- Anda memerlukan model proses yang telah dirancang secara eksplisit untuk mengakomodasi produk yang berkembang dari waktu ke waktu
- Dilakukan iteratif yang memungkinkan Anda untuk mengembangkan versi PL yang semakin lengkap
- Dua jenis diperkenalkan, yaitu **model Prototyping** dan **Model Spiral**

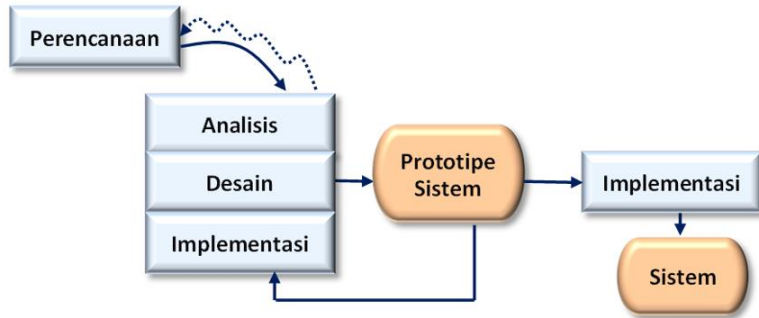


## 2.5.4.a Model Proses Evolusioner: PROTOTYPING

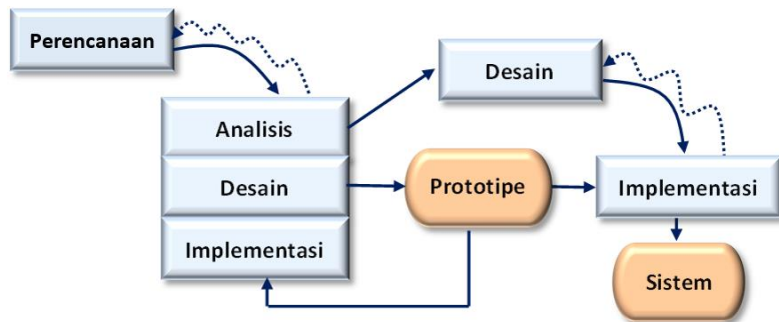
- Prototype dapat dibangun dengan lebih cepat dengan menggunakan fragmen program atau *tool* (misal: *report generator* dan *windows manager*).
- Prototype dapat juga dianggap sebagai "*the first system*" yang pada akhirnya tidak akan digunakan, atau disebut "*throw away*".
- Namun, karena prototype terus diperbaiki sesuai iterasinya, maka pada akhirnya secara **EVOLUSIONER PROTOTYPE** akan menjadi sistem yang sebenarnya (*actual system*).



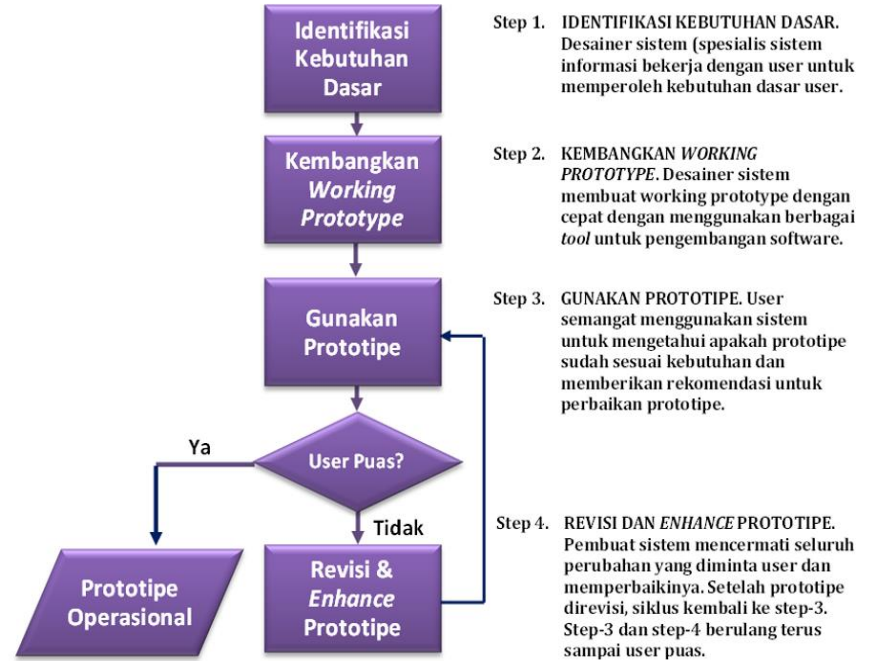
Gambar 2.10 Prototipe Pengembangan aplikasi dengan prototyping, dimana prototyping mengkombinasikan tahapan dalam SDLC (O'Brien & Marakas, 2011)



Gambar 2.11 Prototipe Sistem menurut Dennis *et all.* (2012)



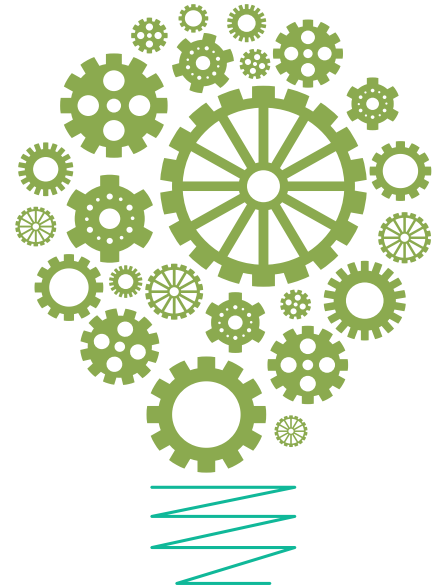
Gambar 2.13 Throwaway prototyping menurut Dennis *et all.* (2012)



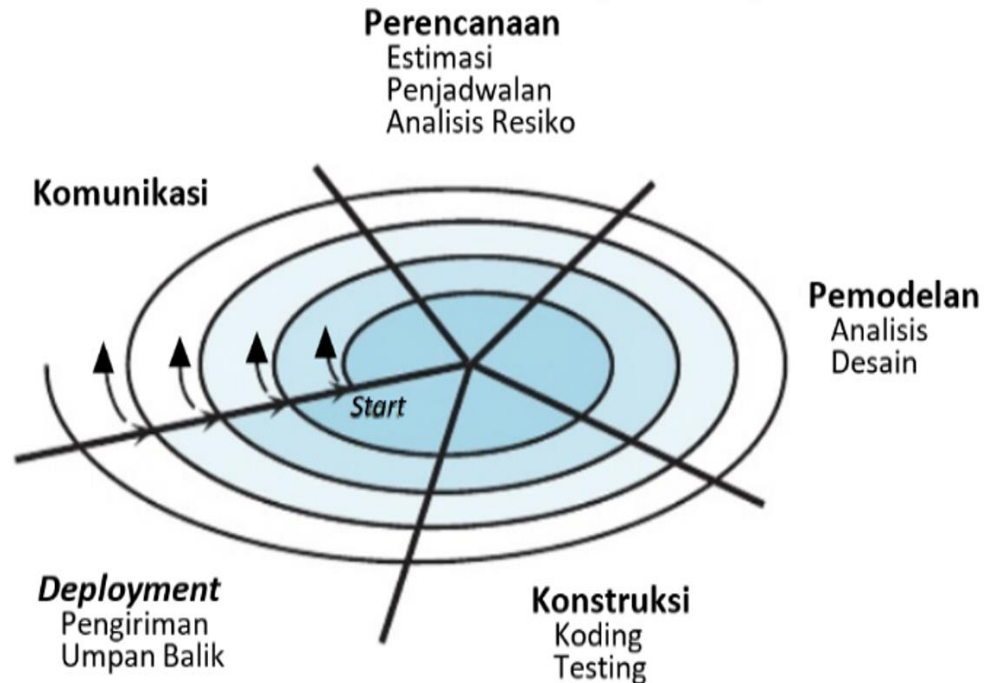
Gambar 2.12 Prototipe Sistem menurut Laudon & Laudon (2014)

## 2.5.4.b Model Proses Evolusioner: SPIRAL MODEL

- Menggabungkan sifat iteratif prototyping dengan aspek terkontrol dan sistematis dari model Waterfall dan merupakan generator model proses yang digerakkan oleh risiko yang digunakan untuk memandu rekayasa serentak multi-stakeholder dari sistem intensif PL.
- Dua fitur pembeda utama:
  1. **Pendekatan siklus** untuk secara bertahap menumbuhkan tingkat definisi dan implementasi sistem sambil mengurangi tingkat risikonya.
  2. Serangkaian **anchor point milestone** untuk memastikan komitmen stakeholder terhadap solusi sistem yang layak dan saling memuaskan
- Serangkaian rilis evolusi disampaikan. Selama iterasi awal, rilis mungkin berupa model atau prototipe. Selama iterasi selanjutnya, versi yang lebih lengkap dari sistem rekayasa diproduksi



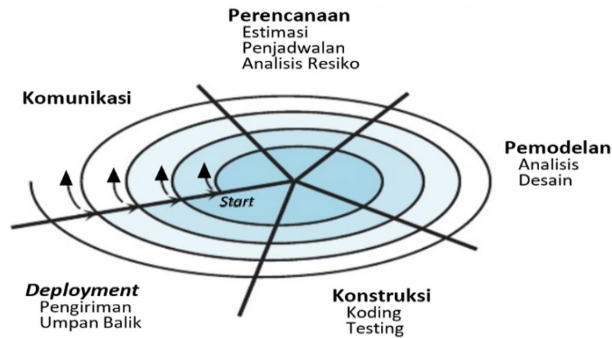
## 2.5.4.b Model Proses Evolusioner: SPIRAL MODEL



- Sirkuit pertama searah jarum jam mungkin menghasilkan **spesifikasi produk**;
- Lintasan berikutnya di sekitar spiral dapat digunakan untuk mengembangkan **prototipe** dan kemudian versi **perangkat lunak** yang semakin canggih.
- Setiap lulus menghasilkan penyesuaian terhadap rencana proyek.
- Biaya dan jadwal disesuaikan berdasarkan umpan balik.
- Juga, jumlah iterasi akan disesuaikan oleh manajer proyek



## 2.5.4.b Model Proses Evolusioner: SPIRAL MODEL



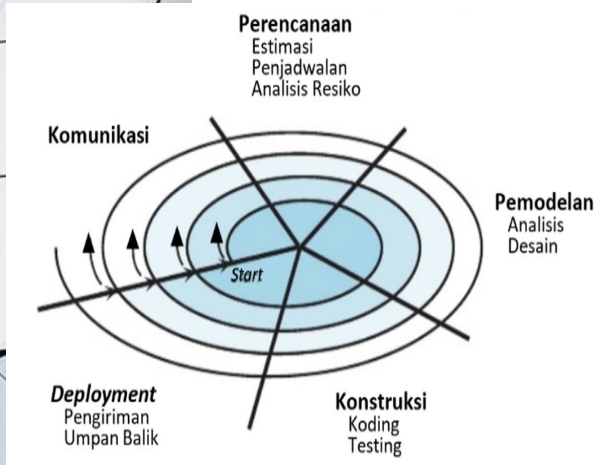
Baik untuk mengembangkan sistem skala besar saat PL berkembang seiring dengan adanya kemajuan proses dan risiko harus dipahami dan ditanggapi dengan benar. Prototyping digunakan untuk mengurangi risiko



Namun, mungkin sulit untuk meyakinkan pelanggan bahwa itu dapat dikontrol karena menuntut keahlian penilaian risiko yang cukup besar

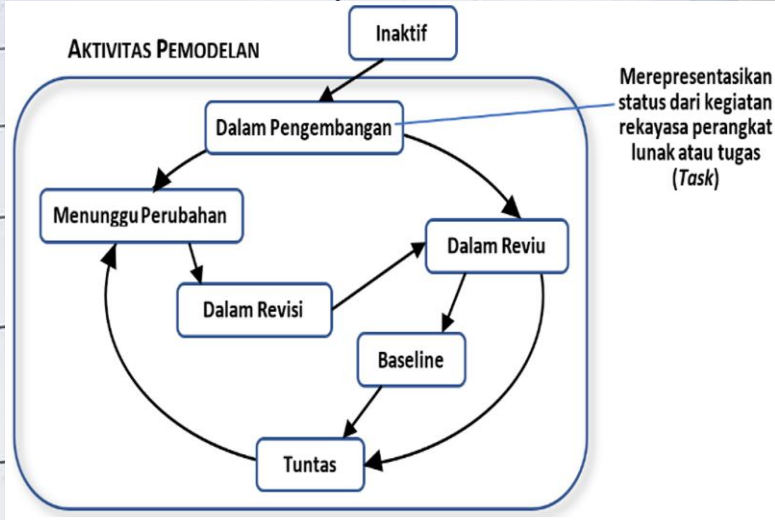
## Tiga Kekhawatiran tentang Proses Evolusi

- 1) Pembuatan prototipe menimbulkan masalah bagi perencanaan proyek karena jumlah siklus yang tidak pasti yang diperlukan untuk membangun produk
- 2) Tidak menetapkan kecepatan maksimum evolusi. Jika evolusi terjadi terlalu cepat, tanpa periode relaksasi, dapat dipastikan proses tersebut akan jatuh ke dalam chaos. Di sisi lain jika kecepatannya terlalu lambat maka produktivitas bisa terpengaruh
- 3) Proses PL harus difokuskan pada fleksibilitas dan ekstensibilitas daripada kualitas tinggi. Kita harus memprioritaskan kecepatan pengembangan di atas nol cacat. Memperluas pengembangan untuk mencapai kualitas tinggi dapat mengakibatkan keterlambatan pengiriman produk ketika peluang yang muncul telah menghilang





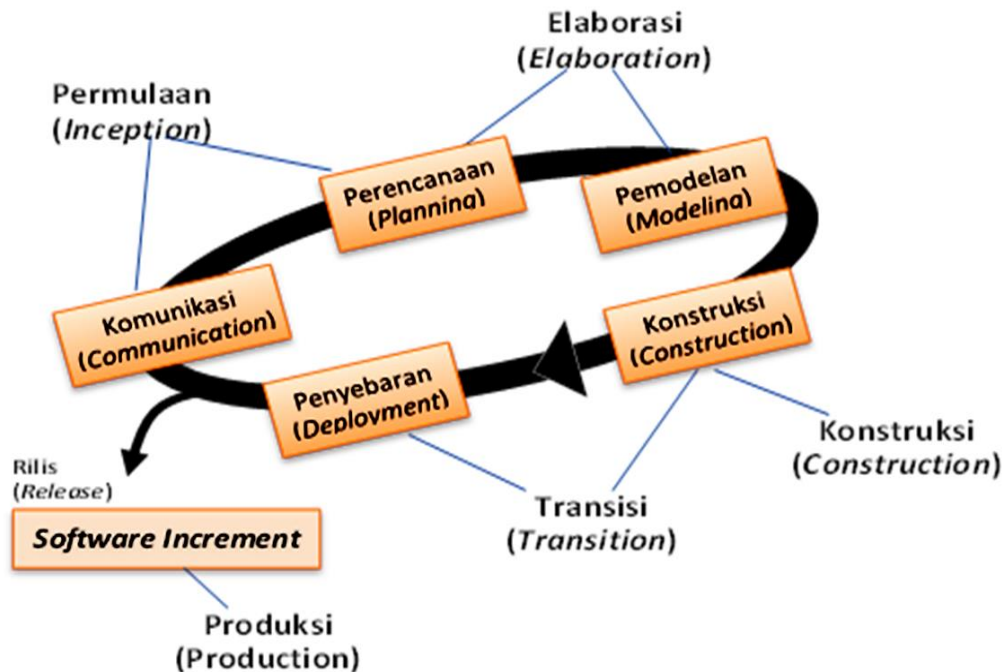
## 2.5.5 Model Concurrent



- ❖ Model konkuren memungkinkan tim PL untuk melakukan elemen iteratif dan konkuren dari salah satu model proses.
  - Misalnya, aktivitas pemodelan yang ditentukan untuk model spiral diselesaikan dengan menerapkan satu atau lebih action berikut: pembuatan prototipe, analisis, dan desain
- ❖ Gambar di samping menunjukkan pemodelan pada salah satu *state* pada waktu tertentu. Misalnya,
  - aktivitas komunikasi telah menyelesaikan iterasi pertama dan dalam *state* menunggu perubahan.
  - Aktivitas pemodelan berada dalam *state* inaktif, dan sekarang membuat transisi ke *state* pengembangan.
  - Jika konsumen menunjukkan perubahan dalam kebutuhan, aktivitas pemodelan bergerak dari *state* pengembangan yang sedang dikembangkan ke *state* menunggu perubahan

- **Model konkuren dapat diaplikasi pada seluruh tipe pengembangan PL dan memberikan gambaran akurat dari status terkini pada proyek.**
- **Model konkuren mendefinisikan jaringan proses, jadi tidak membatasi kegiatan RPL, actions dan tasks ke urutan events.**
- **Setiap aktivitas, aksi, atau tugas pada jaringan ada bersamaan dengan aktivitas-aktivitas, actions dan tasks lainnya.**
  - **Events yang dihasilkan pada satu titik akan memicu transisi di antara state**

## 2.5.6 Model Proses Unified (Unified Process-UP)



Unified Process (UP) adalah proses perangkat lunak ‘*use-case driven*, *architecture-centric*, *iteratif*, dan *incremental*’ yang selaras dengan *Unified Modeling Language* (UML) untuk memodelkan dan mengembangkan sistem berorientasi objek secara iteratif dan *incremental*

UP menerapkan fase-fase *aktivitas generik*, yaitu komunikasi, perencanaan, pemodelan, konstruksi, dan *delivery*.

# Work Products UP

## Fase Awal

- Dokumen Visi
- Modal Use-Case awal
- Glossary Proyek Awal
- Case Bisnis Awal
- Penilaian Resiko Awal
- Rencana Proyek, Fase-fase, dan Iterasi
- Model Bisnis, kalau diperlukan
- Satu atau lebih Prototipe1

## Fase Elaborasi

- Model Use-Case
- Tambahan Kebutuhan, termasuk Kebutuhan Non Fungsional
- Model Analisis
- Deskripsi Arsitektur Perangkat Lunak
- Prototipe Arsitektur yang Dapat Dieksekusi
- Model Desain Pendahuluan
- Daftar Risiko yang Direvisi
- Rencana Proyek, termasuk Rencana Iterasi, alur kerja milestone yang sudah diadaptasi, dan teknis Produk Kerja
- User Manual Awal

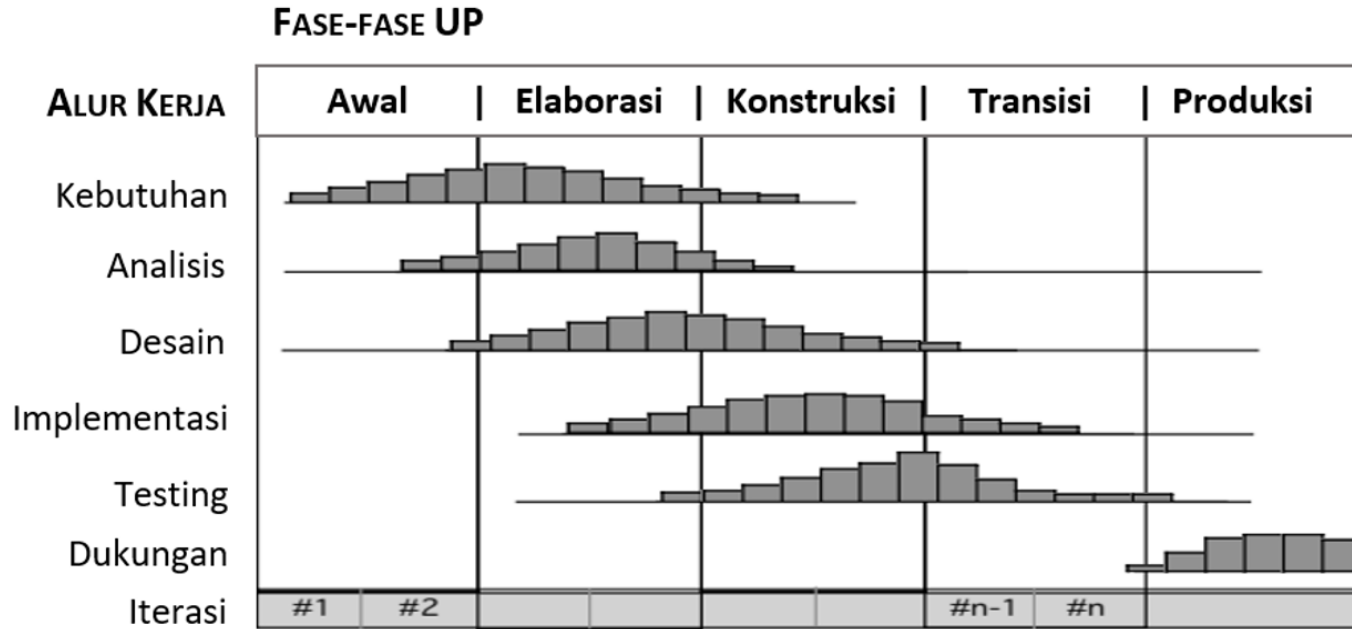
## Fase Konstruksi

- Model Desain
- Perangkat lunak Komponen
- Perangkat lunak Increment Terintegrasi
- Rencana dan Prosedur Testing
- Test Cases
- Dokumentasi Pendukung: User Manual, Manual Instalasi, Deskripsi Increment Saat Ini

## Fase Transisi

- Perangkat Lunak Increment yang diserahkan
- Laporan Beta Test
- Umpan Balik Pengguna secara Umum

# Fase-fase UP dan alur kerja secara *staggered concurrency*



Ada kemungkinan bahwa pada saat fase konstruksi, fase transisi, dan fase produksi dilaksanakan ternyata pekerjaan untuk perangkat lunak *increment* berikutnya sudah dimulai. Hal ini berarti bahwa fase-fase UP tidak terjadi secara berurutan namun secara bersamaan (*staggered concurrency*).



## 2.6 Produk dan Proses

# Kekuatan dan Kelemahan Model Proses

- Pada kenyataannya tidak ada proses yang sempurna untuk setiap proyek
- Biasanya tim PL mengadaptasi satu atau lebih model proses yang dibahas dalam 2.5 atau model proses agile yang akan dibahas di Pertemuan 3 untuk memenuhi kebutuhan mereka akan proyek yang ada.
- Jika prosesnya lemah, produk akhir pasti akan menderita.
- Tetapi ketergantungan yang berlebihan pada proses juga berbahaya.
- Orang memperoleh (atau lebih) kepuasan dari proses kreatif seperti yang mereka lakukan dari produk akhir
- Sebagai seorang profesional perangkat lunak yang kreatif, Anda juga harus mendapatkan kepuasan dari prosesnya seperti halnya produk akhir.
- Dualitas produk dan proses merupakan salah satu elemen penting dalam menjaga orang-orang kreatif tetap terlibat karena rekayasa perangkat lunak terus berkembang



## 2.7 Rangkuman





Model Proses	PROS	CONS
WATERFALL	<ul style="list-style-type: none"><li>❖ Mudah dipahami dan direncanakan</li><li>❖ Berfungsi untuk proyek kecil yang dipahami dengan baik</li><li>❖ Analisis dan pengujian sangat mudah</li></ul>	<ul style="list-style-type: none"><li>❑ Itu tidak mengakomodasi perubahan dengan baik.</li><li>❑ Pengujian terjadi di akhir proses</li><li>❑ Persetujuan pelanggan ada di akhir</li></ul>
PROTOTYPING	<ul style="list-style-type: none"><li>❖ Ada pengurangan dampak perubahan requirements</li><li>❖ Pelanggan terlibat lebih awal dan sering</li><li>❖ Bekerja dengan baik untuk proyek-proyek kecil</li><li>❖ Kemungkinan penolakan produk berkurang</li></ul>	<ul style="list-style-type: none"><li>❑ Keterlibatan pelanggan dapat menyebabkan penundaan</li><li>❑ Mungkin ada godaan untuk "mengirimkan" prototype</li><li>❑ Pekerjaan hilang dalam throwaway prototipe</li><li>❑ Sulit untuk merencanakan dan mengelola</li></ul>





Model Proses	PROS	CONS
SPIRAL	<ul style="list-style-type: none"><li>❖ Ada keterlibatan pelanggan yang berkelanjutan</li><li>❖ Risiko pengembangan dikelola</li><li>❖ Sangat cocok untuk proyek besar dan kompleks</li><li>❖ Bekerja dengan baik untuk produk yang dapat diperluas</li></ul>	<ul style="list-style-type: none"><li>❑ Kegagalan analisis risiko dapat merusak proyek</li><li>❑ Proyek mungkin sulit untuk dikelola</li><li>❑ Membutuhkan tim pengembang yang ahli</li></ul>
UNIFIED PROCESS	<ul style="list-style-type: none"><li>❖ Dokumentasi kualitas ditekankan</li><li>❖ Ada keterlibatan pelanggan yang berkelanjutan</li><li>❖ Mengakomodasi perubahan kebutuhan</li><li>❖ Bekerja dengan baik untuk pemeliharaan proyek</li></ul>	<ul style="list-style-type: none"><li>❑ Use Case tidak selalu akurat</li><li>❑ Memiliki integrasi peningkatan perangkat lunak yang rumit</li><li>❑ Fase yang tumpang tindih dapat menyebabkan masalah</li><li>❑ Membutuhkan tim pengembang yang ahli</li></ul>

# Rangkuman



## Model Proses Generik



Mencakup set dari framework dan umbrella activities, actions, dan set work tasks



Masing-masing dari berbagai model proses dapat dijelaskan dengan aliran proses yang berbeda—deskripsi tentang bagaimana framework activities, actions, dan tasks diatur secara berurutan dan kronologis.



Pola proses dapat digunakan untuk memecahkan masalah umum yang dihadapi sebagai bagian dari proses perangkat lunak

# Rangkuman



## Model Proses Preskriptif



Telah diterapkan selama bertahun-tahun dalam upaya untuk membawa ketertiban dan struktur untuk pengembangan perangkat lunak



Masing-masing model ini menyarankan aliran proses yang agak berbeda, tetapi semuanya melakukan rangkaian aktivitas kerangka umum yang sama:

komunikasi, perencanaan, pemodelan, konstruksi, dan penerapan (deployment)

# Rangkuman

## Model Proses Sekuensial



Waterfall adalah paradigma rekayasa perangkat lunak tertua



Menyarankan aliran proses linier yang sering tidak konsisten dengan realitas modern (misalnya akibat perubahan terus-menerus, sistem yang berkembang, garis waktu yang ketat) di dunia perangkat lunak



Memiliki penerapan dalam situasi di mana kebutuhan (*requirements*) didefinisikan dengan baik dan stabil

# Rangkuman

## Model Proses Incremental



Bersifat iteratif dan menghasilkan versi PL yang berfungsi dengan cukup cepat



Model proses evolusioner mengenali sifat iteratif, inkremental dari sebagian besar proyek RPL dan dirancang untuk mengakomodasi perubahan



Model evolusioner, seperti pembuatan prototipe dan model spiral, menghasilkan produk kerja tambahan (atau versi PL yang berfungsi) dengan cepat



Model-model ini dapat diadopsi untuk diterapkan di semua aktivitas RPL—mulai dari pengembangan konsep hingga pemeliharaan sistem jangka panjang

# Rangkuman

## Unified Process



Proses perangkat lunak "use case-driven, arsitektur-sentris, berulang, dan inkremental" yang dirancang sebagai framework untuk metode dan tools UML



Menerapkan aktivitas generik: komunikasi, perencanaan, pemodelan, konstruksi, dan deployment



Fase-fase UP: Inception (komunikasi dan perencanaan), Elaborasi (perencanaan dan pemodelan), Konstruksi, Transisi (konstruksi dan deployment), serta production



**Pertemuan 2:**

# **Process Model**

**Meuthia Rachmaniah**

**Departemen Ilmu Komputer, FMIPA IPB**

*Software Engineering: A Practitioner's Approach, Ninth Ed.*

*Roger S. Pressman dan Bruce R. Maxim*

Copyright © 2020 McGraw-Hill Education